ADVANCED CONTROL SYSTEM FOR STAND-ALONE DIESEL ENGINE DRIVEN-PERMANENT MAGNET GENERATOR SETS

YANTING HU

Faculty of Computing Science and Engineering

De Montfort University

A thesis submitted in partial fulfilment of the requirements for De Montfort University for the Degree of Doctor of Philosophy (PhD)

July 2001

ABSTRACT

The main focus is on the development of an advanced control system for variable speed stand-alone diesel engine driven generator systems.

An extensive literature survey reviews the historical development and previous relevant research work in the fields of diesel engines, electrical machines, power electronic converters, power and electronic systems. Models are developed for each subsystem from mathematical derivations with necessary simplifications made to reduce complexity while retaining the required accuracy. Initially system performance is investigated using simulation models in Matlab/Simulink.

The AC/DC/AC power electronic conversion system used employs a voltage controlled dc link. The ac voltage is maintained at constant magnitude and frequency by using a dc-dc converter and a fixed modulation ratio VSI PWM inverter. The DC chopper provides fast control of the output voltage by dealing efficiently with transient conditions.

A Variable Speed Fuzzy Logic Core (VSFLC) controller is combined with a classical control method to produce a novel hybrid controller. This provides an innovative variable speed control that responds to both load and speed changes. A new power balance based control strategy is proposed and implemented in the speed controller.

Subsequently a novel overall control strategy is proposed to co-ordinate the hybrid variable speed controller and chopper controller to provide overall control for both fast and slow variations of system operating conditions.

The control system is developed and implemented in hardware using Xilinx Foundation Express. The VHDL code for the complete control system design is developed and the designs are synthesised and analysed within the Xilinx environment. The controllers are implemented with XC95108-PC84 and XC4010-PC84 to provide a compact and cheap control system. A prototype experimental system is described and test results are obtained that show the combined control strategy to be very effective. The research work makes contributions in the areas of automatic control systems for diesel engine generator sets and CPLD/FPGA application that will benefit manufacturers and consumers.

This thesis is dedicated to my parents, Tongzhang HU and Jieru GAO, and my family.

Acknowledgements

I would like to thank:

To my supervisors, for their help and support throughout this research. I am especially grateful to Professor Malcolm McCormick for his invaluable advice, guidance and encouragement and grateful to Dr. M. Cirstea and Dr. P. Urwin for advice and useful discussions.

To the Technical and Administrative staff of Faculty of Computing Science and Engineering, De Montfort University. To Dilip Chauhan, Manbir Sambi, Tim O'Mara, Pritesh Karia and Stephen Regester many thanks are due for their help in various aspects of the project.

To Mrs. Sheila Hayto and Mrs Veena Vora, secretaries of the faculty, for their help and advice proven to be an invaluable asset in many situations.

To the Faculty of Computing Science and Engineering, De Montfort University and the EPSRC for providing the facilities and financial support.

I must specially thank my husband Zhe Chen for all his invaluable technical advice, encouragement and support throughout the period.

I also acknowledge the technical discussions from all my colleagues in the Electrical Machine Drive and Control Group, especially Andrei Dinu, Jean Khor and Abdulmagid Aounis. I also wish to thank Professor E. Spooner, Dr. J. Bumby and other colleagues in Power and Control Group, University of Durham, Professor Lawrance Haydock and other staff of Newage International Ltd. for technical discussions.

Last, but not anyway least, I am particularly grateful to my parents for their unfailing support and encouragement, and again to Zhe, my husband for his support, endurance and help on the thesis preparation; to Cong, my son, for his support and endurance. It is to them that I owe the most.

Contents

Chapter 1

Introduction

1.1. Background and project description

Diesel engine generator systems have been used in electricity generation for about one century. They can be operated in a stand-alone mode or in the grid. In a system where the generators are directly connected to the interconnected grid, the speed of the generators is fixed by the grid frequency, whereas in a stand-alone system, the terminal voltage tends to vary with changing load. An Automatic Voltage Regulator (AVR) is normally used in a synchronous generator to control the voltage at the terminals of the generator and hence that of the power system. However, with a permanent magnet generator, a power electronic interface needs to be used to regulate the voltage. A proposed power electronic interfaced diesel engine generator system is shown in Figure 1.1.

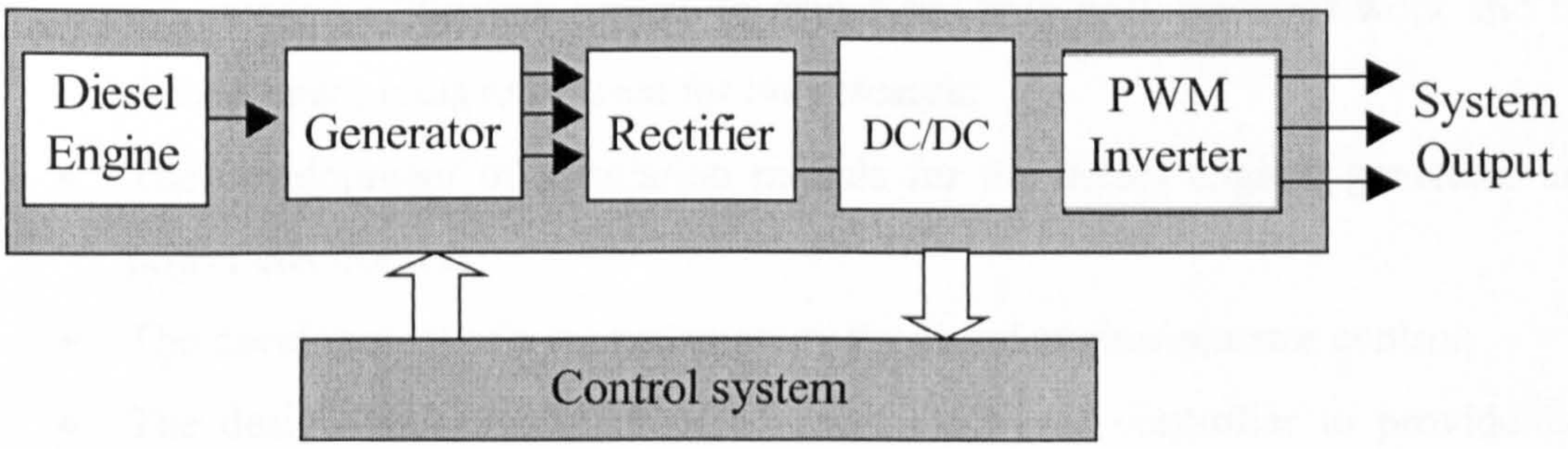


Figure 1.1. Schematic of a stand alone diesel engine generator system

In the system, the diesel engine is a prime mover, which drives the synchronous generator. The ac output of the generator is converted into dc voltage by means of a rectifier. The combination of rectifier and dc-dc converter appears as a voltage source to the inverter, which converts the dc voltage into ac voltage again. The dc link voltage may be regulated by adjusting the generator speed and using a variable ratio chopper. The main function of the Voltage Source Inverter (VSI) is to produce three-phase ac

voltage. In the system shown in Figure 1.1, the engine-generator side frequency is de-coupled from the frequency at load side, so the diesel engine generator can be operated at variable speed.

The control system plays an important role in such a stand-alone generation system in order to ensure the stable and efficient operation of the system. It includes at least two main functions: 1) to maintain a constant output voltage and frequency of the ac system; 2) to regulate the fuel input to balance the power of the system,

1.2. The aims and objectives of the research project

The programme aims to investigate, develop and implement an intelligent control system for a stand-alone variable speed diesel driven permanent magnet generator system. The control system will enable the system to provide an ac output at fixed frequency and voltage amplitude under varying load conditions and to meet the loading demands of consumers by adjusting the fuel rate. The main objectives of this research programme include:

- An extensive literature review to gain familiarity with previous work and the development tools to be used for the research;
- The development of simulation models for the diesel engine, generator and power electronics;
- The development of a control strategy for diesel engine/actuator control;
- The design and simulation of a power electronic controller to provide fast frequency and voltage control;
- The implementation of the developed control strategy into Complex Programmable Logic devices (CPLDs) and Field Programmable Gate Arrays (FPGAs) using Very High Speed Integrated Circuit Hardware Description Language (VHDL) & Xilinx tools/libraries;
- The development and testing of a complete prototype experimental system.

1.3. Methodology

The aims of the project are to be achieved by analysis, numeric computation, time domain simulation and laboratory experiment.

The research methodology adopted is a "top-down" approach. The study and analysis of the characteristics and requirements of the system and the modelling and simulation at the top level are initially performed, followed by a detailed development, simulation and implementation of subsystems and specific circuits elements. Then the final synthesis of the VHDL code, silicon implementation is achieved. Experiments on a small motor generator test system are finally performed to validate the operational characteristics imposed by the proposed control strategy.

With the increasingly competitive nature of modern industry, the required development speed of new products is rapidly increasing. To match the pace, computer aided computation, analysis, design and simulation techniques are adopted. The advantages are:

- a more thorough verification of the design using simulation tools is enabled, thus allowing design optimisation before implementation;
- some of the design steps and design data management aspects can be automated.

The numerical analysis, computation and time domain simulation studies are mainly performed with MATLAB/SIMULINK software package. Xilinx Foundation Express software package is also used for control system simulation study and CPLD/FPGA synthesis.

The control systems are developed on CPLD/FPGA devices with interface system based on the analogue-digital electronic circuitry. The experimental system (power converters with associated driving and control circuits) has been designed, built and tested and provides results for the validation of system simulations.

Chapter 2

Research Background and Historical Review

This chapter briefly reviews the development of engines, electrical machines, power systems, power electronics, the features of small-scale power systems and the applications of diesel engines in power systems. In addition Field Programmable Gate Array (FPGA) technology and the VHSIC Hardware Description Language (VHDL) are reviewed particularly related to the research project.

2.1. History of engine and diesel engine

2.1.1. Engine development history

Engines, the powerful tools for energy conversion, established their respected place in advancing civilisation during the industrial revolution. They are still widely used in modern industry and play a very important role in our daily life. The development of engines has a rather long history and is only briefly dealt with here to give an appropriate background to the subject of this thesis.

In 1765 James Watt of Scotland produced a successful steam engine. Although others had worked on steam engines earlier, Watt's engine was much more efficient. It mainly consisted of three parts: boiler, cylinder with piston and condenser [15]. The steam produced by the boiler is let into the cylinder to push up the piston. The exhausted steam is then passed into the condenser where it is condensed into water. Meanwhile the vacuum created by the steam leaving the cylinder acts to pull the piston down. This cycle is repeated and by this means thermal energy is converted into mechanical energy. Watt's engine was first used to power water pumps in the coal-mines of England. Through the years Watt's steam engine became an extremely efficient and refined machine and was the key to rapid industrial growth. The steam

engine found many applications in industrial and motive applications for example in cotton mills, steamships and railway systems in many countries.

Another type of steam engine, the turbine, was developed towards the end of the nineteenth century. In this type of engine, a jet of steam is directed against the turbine blades and rotates the blades to produce power in a very similar way with that of a windmill. The steam turbine has been mainly applied to power large ships and producing electrical power. Today, the steam turbine is still widely used in power generation.

Steam power was the dominant power source until the development of the internal combustion engine. One of the earliest attempts to produce an internal combustion engine was made by Christian Huygens in the seventeenth century [41]. In his engine, a piston was forced down in a cylinder by the explosion of gunpowder. Gunpowder presented problems, so inventors started seeking other ways for the development. In 1860 [41], J.J.E. Lennoir developed a three horsepower (2.2 kilowatt) engine that burned coal gas. Over several years fourteen hundred engines of this type were built and operated in France and England for powering water pumps and printing presses. In 1876 Nikolaus A. Otto patented his silent gas engine [41]. This engine combined Otto's inventive theories and the work of others into a precision piece of machinery. This successful engine firmly established the four-stroke principle on which the majority of today's internal-combustion engines operate. Although it was successful, it was large and awkward, weighing about 1110 pounds per horsepower (670 kilograms per kilowatt). Gottleib Daimler was the first person who realised that a light high-speed engine was needed [41]. By 1886 Gottleib Daimler had developed a four-stroke cycle engine that operated at a much higher speed, 800 rpm, and weighed only 88 pounds per horsepower (50 kilograms per kilowatt).

2.1.2. Diesel engine developments

Working on theoretical lines, Dr. Rudolf Diesel, in Germany, took out an important patent in 1892 setting out the following principle [27], which he considered should govern any improvement in internal combustion engines, namely:

1. Production of the maximum cycle temperature before combustion by the mechanical compression of the air charge.
2. The gradual introduction of a small accurately measured amount of finely-divided fuel into the heated air, therefore no increase of temperature occurs during the injection period but all of the generated heat is immediately carried off by the gaseous expansion process.
3. The introduction of a large quantity of excess air, i.e. much greater than that needed for combustion of the fuel, to remove the excess heat from the cylinder.

The general principles Diesel enunciated were fundamentally those of present day compression-ignition engines.

In 1897, Diesel produced a successful engine, a single-cylinder engine capable of developing 25 horsepower (18.6 kilowatts). The first commercial diesel engine, a two-cylinder, 60 horse-power (44.8 kilowatts) engine, was put into service a year later in St. Louis, Mo. The use of diesel engines spread rapidly. In a few years, thousands of engines were in use. The Augsburg Krupp Company in Essen, Germany, made a single-cylinder vertical engine of a four-cycle type, operating upon the single-acting principle. It was fitted with small auxiliary air and fuel pumps. The cylinder head was provided with an air inlet and an exhaust valve. The output of this 20 hp engine could be regulated either by varying the time during which fuel was injected or by altering the air pressure in the reservoir. The original engine was much improved by better atomisation of the fuel, fitting a governor and using higher compression pressures.

Until about 1927 these engines were primarily of relatively low speed and of comparatively high power suitable for marine and power station purposes, in place of the less efficient steam engines. Such diesel engines include the opposed-piston,

single, and double acting four-cycle, single acting two-cycle and supercharged versions [27].

These engines were designed in many cases for competition with steam engines of that period, so that no particular attention was apparently given to the subject of weight reduction; most designs used for marine purposes appear to follow the general lines of steam engine practice. The marked fuel economy of the low speed Compression-Ignition (C.I.) engine, the absence of the boiler, and the use of liquid fuel that can be more conveniently stored than coal are among the more important advantages the C.I. engine had over its steam engine rival.

The first engines were heavy, weighing as much as 250 pounds per horsepower (152 kilograms per kilowatt). Today, diesel engines are much lighter in weight and are commonly found on all types of trucks, heavy road machinery, railway locomotives, and in automobiles. Modern engines weigh as little as a few pounds per horsepower.

The era of petrol engine development has resulted in the highly efficient engines of today and has demoted the C.I. engine somewhat into the background. The internal combustion engine has taken over many jobs of the steam engine and has found countless applications. The smaller, internal combustion engine was ideally suited for land vehicles and led to the development of automobiles, trucks, tractors, etc. However it has been recognised in more recent years that the C.I. engine has certain marked advantages particularly of fuel economy, simplicity and safety (from the inflammable fuel point of view). Much attention has therefore been given to improving its design so that it can successfully compete with the petrol engine in the various fields of application.

After about 1940 attention was given to methods of increasing C.I. engine speeds, reducing their specific weight, and improving their thermal efficiency. It was realised that it was be necessary to depart from the constant pressure cycle, since the constant pressure cycle was found to be more suited to lower speed engines. In addition the air injection method was replaced by a mechanical system. In this connection, valuable

pioneering work was carried out on a single-cylinder engine converted from a petrol engine at the Royal Aircraft Establishment at Farnborough under H.B. Taylor [27].

Diesel engines are all high compression engines and the high compression ratio gives the diesel engine an efficiency advantage. The simple steam engine has an efficiency of 6 percent to 8 percent. The steam turbine and condensing engine have an efficiency of 16 percent to 33 percent. The gasoline engine has an efficiency of 25 percent to 32 percent. Of the common engine types, the diesel engine is the most efficient. The diesel engine has an efficiency of 32 to 38 percent.

Nowadays, engine speed has increased from a few hundred rpm to as much as 5000 rpm in small output engines, however the ordinary speed range for the engines is about 1500 to 4500 rpm.

2.2. Electric machines and power systems

2.2.1. Electric machines

Development of the electric machine started in early nineteenth century. The early examples of the application of electric current to produce mechanical power date back to the time following Faraday's fundamental discoveries (1821) on electromagnetic rotation. Professor Dal Negro, of Padua University, produced the first electric motor in the year 1830 and obtained rotary motion from current supplied by a voltaic battery [15]. Four years later, Jacobi, a Russian professor, started a series of experiments that led to the first electrically propelled boat [15]. In 1840 Robert Davidson of Scotland equipped a car with an electric motor which consisted of eight coils on cylinders of wood with sliding cores of soft iron acting like engine pistons and connected to cranks on the car axles [15]. During the decades before and after 1890, inventors' attention was given to developing the dc machine. To accommodate different requirements, dc machines operating over a range of speeds and having various shapes to fit in with items of other equipment became available, and very soon established themselves for industrial applications.

The development of ac electric machines started around 1880s. The first practical type of an ac machine was the induction motor, which became a valuable part of modern electrical equipment. Arago produced a rotating field by spinning a permanent magnet, and the rotor, a copper disc, was dragged round after it, or vice versa, through the interaction of the induced eddy current in the disc and the field of the magnet [15]. The growth of the ac machines was further advanced by Nikola Tesla who presented a paper describing two-phase induction and synchronous motors at a meeting of the American Institute of Electrical Engineers in 1888 [15]. Since the invention of the electric machine, both dc and ac machines have rapidly developed. Nowadays, electric machines are widely used.

There are three main classes of generators: dc, ac synchronous and ac induction. The basic features of each type of generator are outlined below.

DC generators

The stator consists of a number of poles, the field windings of which are excited by dc current. The rotor consists of conductors wound on an iron armature which are connected to a commutator. Electric power is extracted via brushes, which ride on the commutator. Direct current machines generally require regular maintenance and are expensive.

Nowadays, for most dc applications, for example battery charging, it is more common to employ an ac generator (alternator), frequently permanent-magnet excited, to generate ac voltage, which is then converted to dc voltage by using simple solid-state rectifiers.

AC synchronous generators

In a synchronous generator, the rotor is excited with a dc current and is driven to rotate, then a 3-phase voltage is generated in the stator at a frequency determined by the speed of rotation.

When connected to a grid, an important feature of the synchronous generator is that the rotor speed must exactly match the synchronous speed. Loss of synchronisation will occur if rotor torque becomes too high, and it is therefore important to know the generator's 'pull-out' torque to avoid such eventualities.

The reactive power characteristics of synchronous generators can be controlled and therefore such machines are often used to supply reactive power to effect voltage control of the power system. It is normal for a stand-alone system to incorporate a synchronous generator connected to a mechanical power source, often a diesel engine.

AC induction generators

The induction (asynchronous) generator differs from the synchronous generator in several aspects. A rotating magnetic field is induced by the application of an ac voltage to the stator. Power is generated due to a difference in the frequency of the rotating magnetic field and the speed of the rotor. This characteristic is termed slip and often the speed differences are only one or two percent from the normal design point. Thus when the machine is generating, the rotor does not rotate at synchronous speed, but at some slightly higher speed.

Induction generators are consumers of reactive power, and it is not common for them to operate in isolation from other plant, but this is possible using special power electronics. It is recommended, especially in a small network where the capacity of the other plants to generate reactive power may be limited, that power correction capacitors are added to the induction generator to compensate for the reactive power demand on the system.

2.2.2. Power systems

There are several milestones in the history of electric power systems that significantly pushed forward their development. Initially they operated using direct current.

In 1878, Thomas A. Edison began working on the electric light and formulated the concept of a centrally located power station serving distributed lighting in a

surrounding area. The operation of the historic Pearl Street Station in New York City on 4 September 1882 marked the beginning of the electric utility industry. At Pearl Street, dc generators, then called dynamos, were driven by steam engines to supply an initial load of 30kW for 110V incandescent lighting to 59 customers in a one-square-mile area. The introduction of the practical dc motor by Sprague Electric, as well as the growth of incandescent lighting, promoted the expansion of Edison's dc systems. The three-wire 220V dc system was developed. At about the same time, the Holborn Viaduct Generating Station in London produced a scheme comprising a 60kW generator driven by a horizontal steam engine, the generated voltage being 100V direct current. This was the first power station in Britain to cater for general consumers.

The commercially practical transformer, developed by William Stanley in 1885, provided the ability to transmit power at high voltage with corresponding low current and lower line voltage drops that made ac systems more attractive than dc systems. The first single-phase ac line in the United States operated in 1889 in Oregon, between Oregon City and Portland at a distance of 21 km with an operating voltage 4 kV [40].

The ac machine (two-phase induction and synchronous motors), presented by Nikola Tesla in 1888, made evident the advantages of polyphase versus single-phase systems [40]. The first three-phase line in Germany became operational in 1891, transmitting power for a distance of 179km at 12kV. Early ac systems operated at various frequencies including 25, 50, 60 and 133Hz. Today, the two standard frequencies for generation, transmission, and distribution of electric power in the world are 60Hz (in the United States, Canada, Japan, Brazil, etc.) and 50Hz (in Europe, the former Soviet Republics, China, etc.).

The electric utilities initially operated as isolated systems, but technical developments that included advances in insulation, protection and control, led to wider area networks and the electric utilities were gradually interconnected to operate in parallel. This offers improved reliability and economy.

From the beginning of 1882 to date, the electric utility industry has grown at a remarkable pace—a growth based on continuous reduction in the price of electricity, primarily due to technological accomplishment and creative engineering.

2.3. Stand alone diesel engine generation system

2.3.1 Stand alone power system

Historically, all early power systems were decentralised, i.e. power was produced at the location where it was consumed. The development of grid networks with centralised generating capacity saw the demise of many decentralised power systems. However, today, there are still many locations in the world which do not have an electrical connection to a central utility network. Furthermore, in many of these places, due to remoteness and cost, although the need for power exists, it is unlikely that a main grid connection will ever be established. Therefore power systems, which can generate and supply electricity to such remote locations, are needed and they are variously termed 'remote', 'decentralised', 'autonomous', or 'stand-alone'.

The nature of the electrical load at a decentralised site is of primary importance and depends strongly on the type of electrical apparatus connected. Broadly speaking, there are three types of applications for remote electrical power:

a) Power for specialised applications, standby power supply (for example emergency power supply for hospitals, broadcast systems, communications stations, etc.) and irrigation in remote areas.
b) Power to remote communities in industrialised countries, and on islands.
c) Community power generation in developing countries.

Each has its own particular requirements and design constraints, for example system reliability can be more important than cost of power in unmanned communication stations. Communities in industrialised countries quickly develop high expectations of power quality and availability as well as competitiveness, whilst in developing countries simplicity of maintenance is a prime consideration.

A community load tends to vary in a more or less regular way over the day, often reaching a maximum sometime during normal working hours and falling to a low level in the very early hours of the morning. As well as the gradual changes, there are also fluctuations of much shorter duration caused by switching in or out large electrical equipment. Table 2.1 gives some examples of typical appliance power consumption [36]. There may also be substantial changes in the load with the time of week and with the season of the year.

Table 2.1 Examples of typical appliance power consumption

Appliance	Power Consumption (W)
Air Conditioner	1400-2000
Battery Charger	Up to 800
Refrigerator	600-1000
Microwave Oven	1000-1500
Electric Frying Pan or Wok	1000-1500
Electric Stove Element	350-1000
Electric Water Heater	1000-1500
Electric Iron	500-1200
Electric Hair Dryer	800-1500
Coffee Percolator	550-750
Electric Drill	250-750
Television	200-600
Radio	50-200
Electric Broom	200-500
Electric Blanket	50-200

The loading level of the power system may change significantly during the day but the voltage and frequency of the system have to be kept nearly constant. The automatic voltage regulator (AVR) and governor are used to affect this. Table 2.2 shows typical voltage and frequency disturbances that may be expected [17].

Table 2.2 Variation limits of voltage and frequency

Steady State Voltage ± 0.5%	25% load step (back to steady state limits in 1 second) ± 2.5%
Steady State Frequency ± 0.5%	25% load step (back to steady state limits in 2 second) ± 2.0%

The ratio of any load changing step to the system power capacity in a stand-alone system is significantly larger than that in a power network. Therefore, load changes can cause the system to more readily depart from the desired operating condition, especially in stand-alone systems with small generators of power rated less than 10 kW. To satisfy load-change requirements and maintain the system at the desired operating condition, the control system plays an important role in this type of system to maintain the voltage and the frequency at the required level.

In remote areas, it is often preferred to adopt permanent magnet generators since the maintenance requirements of PM generators are lower than that of the field winding exciting generators. Then voltage control needs to be performed by other methods such as power electronic converters discussed later.

Three main types of prime movers can be found in industrial power systems, namely gas, steam turbines and diesel engines. They all have specific operating parameters which affect their suitability for use in stand alone power system. Gas turbines are available from around 500 kW to about 70 MW in discrete ratings and therefore cannot be exactly tailored to the power demand in an autonomous system. Steam turbines are available in sizes from 1 MW to units in excess of 1500 MW. Consequently steam turbines are mostly used in a large-scale power system where long periods of operation are necessary, and in utility systems where steam is available or is required for an industrial process.

Diesel engine sets are available in unit sizes from 5kW to more than 24MW. Diesel engines are the most flexible prime movers in terms of operation and electrical generating efficiency, and consequently are used in industrial power systems throughout the world. Diesel engine alternator sets are used when the electricity

supply requirement is the only or predominant load requirement or in relatively small power networks such as those associated with island systems, industrial complexes, locomotive and marine systems.

2.3.2. Diesel engine application to power systems

At present, the most common way to supply electricity to remote loads, whether communities or special applications, is with a diesel engine driving a generator set. For small loads a single diesel set might be appropriate, whereas for larger communities multiple diesels are commonly employed. In the latter case, one or more diesels, typically the most efficient, supplies the base load. Frequently, enough reserve capacity is provided so that at least one machine can be taken out for overhaul at any given time.

Multiple diesel grids can run at high efficiency since it is possible to ensure that all running plants are highly loaded. Diesel manufacturers strongly advise operating above a minimum load, typically 40 per cent, in order to minimise engine wear and to maintain high efficiency. However, due to continually changing load patterns and the desire to always meet the demands of the load, many diesel systems are incorrectly sized or inefficiently controlled. This quite often results in the diesels not complying with the operational constraints, including particular emission level, nevertheless, diesel plants offer reliable power supply.

The main disadvantage with diesel electric grids is that the cost of power tends to be high, often many times greater than that for larger capacity networks. Poor economics are traceable to the cost of diesel fuel, including the cost of transportation that is often the dominant factor, efficiencies of operation and the cost of operation and maintenance in a remote location. The major advantage of diesel systems is that they are extremely well proven, and if maintained correctly, highly dependable. The key point is that although diesel electric power is often reliable, it is also expensive. Therefore, any increase in efficiency of the systems is very useful.

A modern diesel engine generator set includes a diesel engine, and a generator, power electronics and their control systems. The diesel engine converts fuel (diesel oil) into mechanical energy and subsequently the generator converts the mechanical energy into electrical energy. Speed regulation and controls are necessary to maintain the desired voltage and frequency to meet the power demand of the system.

2.4. Power electronic developments

The prehistory of modern power electronics began with the introduction of the mercury arc rectifier in 1900. Subsequently the metal tank rectifier, grid-controlled vacuum-tube rectifier, ignition, phanotron, and thyratron were introduced gradually. These devices were applied for power control until the 1950s when the electronics revolution provided more sophisticated devices.

The electronics revolution began in 1948 with the invention of the silicon transistor at Bell Telephone Laboratories by Bardeen, Brattain, and Schockley. Most of today's advanced electronic technologies are traceable to that invention. The next breakthrough in 1956, was the invention of the PNPN triggering transistor at Bell Laboratories, which was defined as a thyristor or Silicon Controlled Rectifier (SCR). Since then, many different types of power semiconductor devices and conversion techniques have been introduced. In addition the microelectronics revolution using silicon semiconductors gave the ability to process a huge amount of information at incredible speed, while the power electronics revolution provides the ability to shape and control large amounts of power with ever-increasing efficiency. The combination of microelectronics with power electronics, has led to many advanced applications of power electronics.

Electric power is used in almost every part of the modern home, commerce and industry. Generation, transmission, distribution and utilization of electric power take place at different levels to suit the available components, processes and technologies involved. Interfacing of different technologies needs intelligent control and conversion techniques. Electric power is the muscle of modern industry and power electronics

makes its utilization smarter. The objective of power electronics is to improve the quality and utilization of electric power. The focus in power electronics is on conversion, efficiency of conversion and control of energy. Recent advances in semiconductor devices have contributed very significantly to research and developments in the field of modern power electronics. Power electronics applications span a wide range on the power scale: milliwatts in wireless personal communication sets or cordless screwdrivers to megawatts in high voltage dc transmission systems and huge industrial motor drives. The power electronics revolution has gained increased momentum since the late 1980s and early 1990s. It is anticipated that power electronics will increasingly shape and condition electrical power at transmission, distribution and consumption levels.

2.4.1. Semiconductor devices

Present power semiconductor devices available for application in power conversion equipment have been developed from the range of diodes, thyristors and transistors into a large family of semiconductor devices. Nowadays, there is a development of modern power electronic devices has resulted in improvement of important characteristics, for example:

1) Decrease in the complication (power level) of the interface between control electronics and power electronics;
2) Decrease in the switching transition times, (this determines switching losses and maximum repetition frequencies);
3) Decrease in conduction voltage drop (losses), (this determines current rating, device cost, process yield, efficiency and cost of cooling requirements);
4) Elimination of the bulky forced commutation components by improved turn-off characteristics of power switching devices.

Commonly used semiconductors include thyristors, asymmetric thyristors, gate-turn off thyristors (GTOs), MOS controlled thyristors (MCTs), integrated gate commutated thyristors (IGCTs), power field-effect transistors (MOSFETs and JFETs), bipolar power transistors and insulated gate bipolar transistors (IGBTs). The switching times for these devices are in the range of 10 ns for some MOSFETs, to a few hundred

microseconds for the largest and slowest thyristors. The voltage ranges from below 100 V to over 4000 V.

Decreased switching times has been achieved by the development of devices, particularly MOSFETs, IGBTs, bipolar power transistors and asymmetric thyristors at different power levels. The short switching times have accentuated the thermal shock and dissipation problems caused by the concentration of the switching losses, so a consequent development in snubbing technology as a means of eliminating the adverse effects of these ultra-fast thermal transients was necessary. Snubbing implies dealing with stored reactive energy at turn-on and turn-off requiring capacitive and inductive reactances to be discharged. However with the increased frequency of operation the dissipation of the reactive energy is impractical, therefore the technology of regenerative snubbing has been developed. This has enabled the development of high frequency power conditioners and converters.

Improvements in interfacing devices, for example, the family of field-effect devices (MOSFETs) and MOS gate devices (IGBTs) have tended to reduce the interface complication. Owing to the gate structure of the devices, charge establishment and removal requirements for turn on and off are less severe than that with bipolar transistors and gate-turn-off thyristors when operating at comparable switching speeds and repetition frequencies.

2.4.2. Power electronic converters

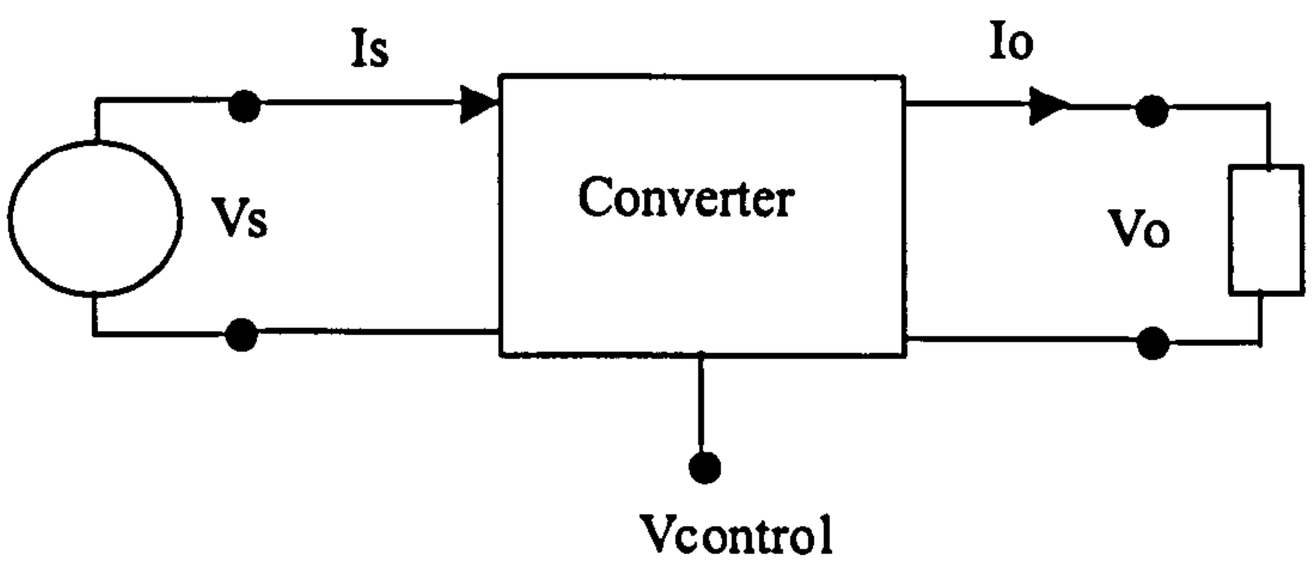


Figure 2.1. Power electronic converter system

The schematic of a power electronic converter system is illustrated in Figure 2.1. The source provides power, the converter converts it into a usable form to suit the sink where the power is utilized. The source is either a dc or an ac voltage/current source. The ac voltage and current waveforms can be either the single-phase or a multiphase (usually three-phase) type. The sink is either an electric load, or another source, or a device where the electrical energy is converted into non-electrical form such as a motor. The converter system is comprised of semiconductors, reactive components and maybe transformers. On the basis of the type of the source and the type of desired output characteristics, power electronic converters may be classified into four categories as shown in Table 2.3:

Table 2.3. Converter classification

Output / Source	DC	AC
AC	Rectification	AC Control
DC	Conversion	Inversion

DC to AC Converter (Inverter)

The dc-ac converter (inverter) takes power from a dc source (voltage or current) and delivers to a load. The output variable of the converter is a single-phase or multi-phase ac voltage or ac current. A practical inverter has either a battery, a solar powered dc voltage source or an ac voltage source (50-60 Hz or variable frequency) derived dc voltage source (often unregulated). Dc-ac converters are widely used in very low-power portable electronic systems such as the flashlight discharge system in a photography camera to very high-power industrial systems. The input dc is converted into an ac of high or low frequency depending on the application, for example 50-60 Hz in emergency lighting systems to high frequency 1MHz in induction heating systems. The output voltage waveform contains rich harmonics. Usually, the desired waveform is a sine wave in low-frequency applications such as the battery powered ac sources for emergency applications. A filter may be used at the output to shape the waveform as desired.

Three-phase ac waveforms are generated by switching switches connected in three parallel legs. A more complex topology consists of a network of switches. The amplitude and frequency of the ac waveform is controlled by controlling the ON-time of switches, the frequency of switching and the sequence of switching. The waveform-shaping filter connected to remove the undesired content of the ac waveform consists of reactive components, inductors and capacitors. The instantaneous difference in energy between the input and output sides is temporarily stored in the filter.

Typical applications of inverters include: aircraft and space power supplies, uninterruptible power supplies, variable-frequency ac motor drives, aircraft variable-speed constant frequency supplies, and induction heating supplies.

AC to DC Converter

The dc-ac and ac-dc converters are functionally similar except for the reverse direction of energy flow. The ac-dc converter (rectifier) takes power from one or more ac voltage/current sources of single or multiple phases and delivers to a dc load. The output variable is a low ripple dc voltage or dc current. Many practical ac-dc converters use a voltage source with line frequency of 50 or 60 Hz, single-phase or three-phase. Ac to dc converters are widely used in applications ranging from very low-power battery charger and dc power supply systems to very high-power dc motor drive systems. Rectifiers are commonly used for dc motor drives, regulated dc power supplies, high voltage dc transmission and wind power generator converters.

The instant of turning on and off semiconductors relative to the ac source waveform, determines the shape of the output voltage waveform. Consequently the dc and ac components of the waveform can be controlled to some extent. L-C filters may be used to remove undesired frequency components.

DC to DC Converter

The dc-dc converter takes power from a dc source (voltage or current) and delivers to a dc load. The output of the converter is one or more low-ripple dc voltages or dc currents (often regulated). The dc output voltage may be different in amplitude from

the input source voltage. This converter can also be thought of as a dc transformer. A practical dc-dc converter has either a battery, a solar powered dc voltage source or a line frequency (50-60 Hertz) derived dc voltage source (often unregulated). The dc-dc converters are used from very low-power portable electronic systems to high-power industrial systems.

Dc-dc converters can be found in applications in electric transportation, high performance regulated power supplies, electronic ballasts and dc motor drive systems.

AC to AC Converter

The ac-ac converter takes power from an ac voltage or current source and the output is an ac voltage or ac current of equal or different frequency than the input ac source. The practical ac converters usually use line frequency 50-60 Hz voltage sources, single or three phase. The ac-ac converter in which the output frequency is lower than the frequency of the source is often called the cycloconverter: The output frequency in a cycloconverter is a fraction of the source frequency. They are used in very high-power industrial applications.

The ac-ac converter with no change of frequency is referred to as the ac controller. The instant of turning on the switch relative to the ac source waveform determines the shape of the output voltage waveform, which in turn determines the rms voltage at the output terminals. The ac-ac conversion can also be achieved by first converting the input ac to the dc and then the dc to the ac of desired frequency, amplitude and number of phases. Such a converter is called the dc-link ac-ac converter. In this type of conversion, there is no restriction on the output frequency relative to the source frequency. Some applications of ac-ac converters include light dimmers, motor speed controls, voltage regulators, electronic tap changers, VAR regulators, and solid-state relays.

Power electronic converters are used over a wide spectrum of industrial applications at different power and frequency levels. A typical application as represented by the Pulse-Width Modulation (PWM) converters, such as the sine wave PWM converters

that have the capacity to improve the performance of converters, have been widely used, and many PWM schemes have been proposed. For example, the sinusoidal modulation generated in a space vector representation [48], the predictive current controller [24], the current feedback controller [33], and neural network based adaptive control strategies, all seek to provide higher performance control in applications such as induction motor drives [48].

A predictive neural current control scheme for a VSI-PWM inverter has been developed by Dinu, et al. [14], [13]. The scheme is based on the state space observer principle and the control is independent of the load parameters. The control strategy does not look into the inside of the system load. It only pays attention on the voltage across the load and the current injected into the load. The on-line inductance estimator (digital neural network) provides an estimated inductance value used for the voltage calculation. In the above scheme, the neural network technique has been adopted for the design of the control structure, so that the control system is compact and responds fast. A four-layer Feed-Forward Artificial Neural Network (FFANN) is used, which has parallel processing capability and the delay time is only the time needed for the signal to pass through the layers of logic gates which implement the FFANN.

2.4.3. Power electronics in diesel engine generation system

Power electronic converters can have a number of useful roles to play in diesel systems. A power electronic system can be placed between the generator and the network, as shown in Figure 2.2. By using an ac-dc-ac conversion in the case of a synchronous or asynchronous generator, or a dc-ac conversion in the case of a dc generator, the system frequency is determined by the load side inverter, therefore, the diesel engine-generator set can be allowed to operate at a variable speed. If power electronic equipment is used to connect to the grid, then the voltage output level of the synchronous generator must be compatible. In the case of a PM generator, the amplitude of the system voltage can be controlled either by an inverter or by a dc-dc converter. By effectively frequency de-coupling the drive from the network, the diesel engine-generator system can be controlled to meet the varying load demands

efficiently. Power electronics can also be used to condition the power supplied from storage devices of significant capacity such as wide range flywheels or battery banks. In fact, power electronics can provide an effective interface between dc or variable frequency power sources and the 'fixed' frequency grid.

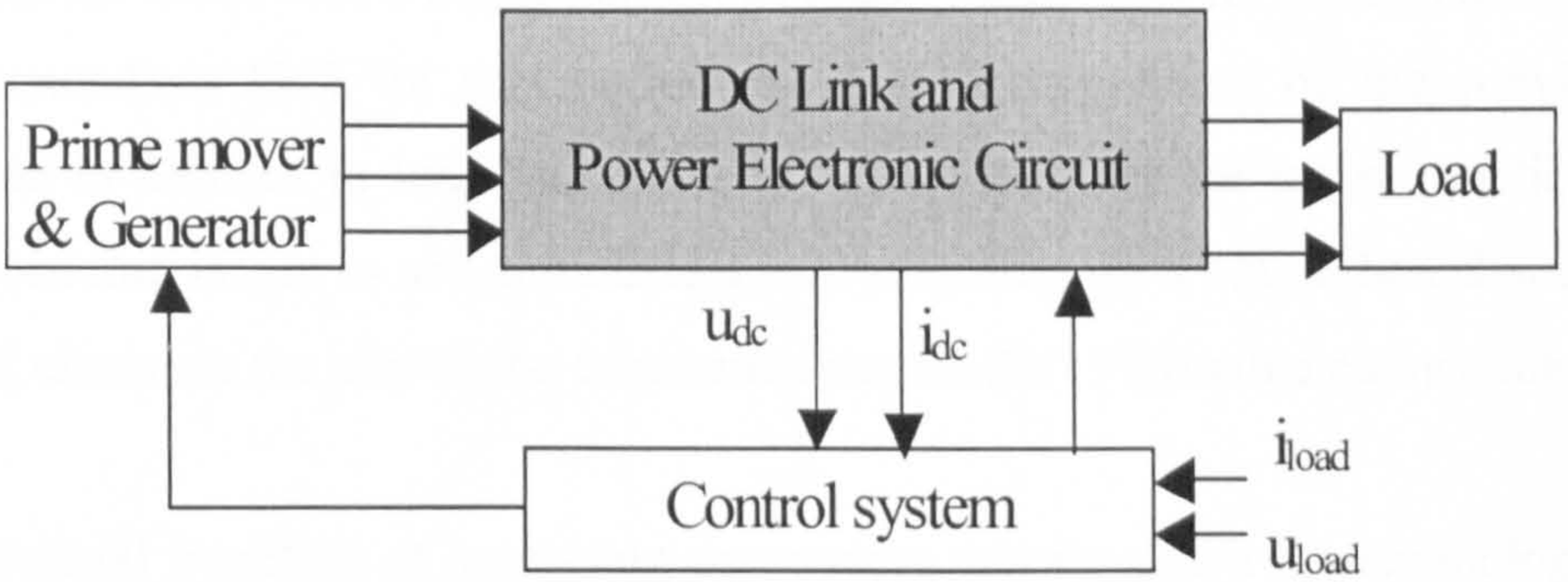


Figure 2.2. Power electronics in diesel-generator system

2.5. VHDL and CPLD/FPGA techniques

2.5.1. CPLD/FPGA development

The Complex Programmable Logic Device (CPLD) and Field Programmable Gate Array (FPGA) are relatively new types of components for the construction of electronic systems, particularly those based on digital, or logical circuit principles. Digital electronics has experienced tremendous changes, moving from the breadboard systems in the 1960s where gates and flip-flops were directly wired together to the newly developed high-density field programmable gate arrays which may consists of more than one million gates.

The early period (the 1960s into the 1970s) was typified by Texas Instruments Transistor-Transistor Logic (TTL) [44], and National Semi-conductor's 4000 series, popularly called Small-Scale Integration (SSI) and Medium-Scale Integration (MSI). Each provides logic in the form of gates, flip-flops, register transfer components, counters, and Arithmetic-Logic Units (ALUs), at the chip level. At the board level, these components were used in relatively small numbers on relatively small boards.

Large-scale integration (LSI) and Very large-scale integration (VLSI) began in the 1970s with the invention of the Dynamic RAM, Intel 1103, and microprocessor, Intel 4004 etc., and evolved to the point where the 32-bit microprocessor became a common component, for example, the Motorola 68000. Technology progress spawned two major market areas with a larger number of market segments, for example, SRAM and micro-controllers [35]. In this period the logic components of the previous era continued to be used in large quantities to "glue" together the larger function chips, boards became larger to accommodate a whole system on a single board and reduce cost and eliminate the pin/wiring constraints imposed by PCB edge connectors.

The industrial intention of increasing component density drove the technology ahead such that by the late 1980s Application Specific Integrated Circuits (ASICs) became available. Mask-programmable gate arrays, cell-based ICs (CBICs), and the first programmable components, programmable array logics (PALs) and field-programmable logic arrays (FPLAs), emerged during this decade as ASIC solutions. Basically there are several varieties of ASIC: gate arrays, Cell-based ICs (CBICs), and programmable logic devices (PLDs). At this stage, boards remained large but delivered more functions, including microprocessor, memories, ASICs.

With the emergence of complex programmable components, especially CPLD/FPGAs, a traditional microprocessor can be complemented by a number of SRAM-programmed FPGAs, memory parts and even a special-purpose processor. The development of CPLD/FPGA is part of the continuing evolution of very-large-scale integrated (VLSI) circuit technology toward denser and faster circuits. The creation of the Programmable Logic Devices (PLDs) can be attributed to Monolithic Momoties, who introduced the first PLA in 1978. By late 1980s the conceptually simple PLDs are complemented by the emergence of Complex Programmable Logic Devices (CPLDs) [9]. At the same period Field Programmable Gate Arrays (FPGAs) were further developed. The two key commercial examples of FPGA device are the Xilinx and Actel FPGAs [43]. The Xilinx FPGA architecture was designed by Ross Freeman and introduced in a paper presented at the 1986 Custom Integrated Circuits Conference [8].

The Actel FPGA architecture was designed by a team led by Amr Mohsen and EI Gamal, and was described in a series of papers early in 1988 [8][16].

In recent years, CPLD/FPGAs have quickly developed and are having a significant impact. With the ability to reconfigure an electronic component, either to correct an error or to change an application, the CPLD/FPGA has economic benefits. In addition with its programmability and adaptability, the CPLD/FPGA has significant advantages especially for the development of prototype systems and their rapid introduction to the market.

2.5.2. The VHDL language

VHDL is a double acronym, namely V stands for VHSIC (Very High Speed Integrated Circuit), and HDL stands for Hardware Description Language. As an implementation independent method of describing electronic systems, VHDL is one of a few HDLs in widespread use today. The language was developed in the early 1980s. In 1983, the United States Department of Defense sponsored the development of the Very High Speed Integrated Circuit (VHSIC) hardware description language (VHDL). The original intent of the language was to serve as a means of communicating designs from one contractor to another in the VHSIC program. However, the design of the language has received input from many individuals in the computer industry and thus reflects a consensus of opinion as to the characteristics a hardware description language should have. In August 1985, version 7.2 of the language was released by the United States Department of Defense (MIL-STD-454L), representing the completion of the first major stage of the language development. Version 7.2 was a complete language, which comprehensively provided constructs for structural and behavioral modeling, as well as a means to document designs. After the release of version 7.2, the Institute of Electrical and Electronics Engineers (IEEE) sponsored the further development of VHDL, the ultimate goal being the development of an improved standard version of the language. The review process was completed by May 1987 and the language reference manual (LRM) was released for industrial review. In June 1987 the eligible IEEE members voted to accept this version of VHDL as the standard version and in

December 1987 (IEEE Standard 1076, 1987), was officially adopted by the IEEE [2]. The IEEE requires that all of its standards are regularly reviewed, and a new version was defined in 1993, to add some new capabilities, remove some ambiguities, while remaining upwardly compatible.

2.5.3. The VHDL language in use with CPLD/FPGA

As a standard language for documenting/defining hardware, VHDL can be used to describe the architecture and behaviour of discrete electronic systems, and the function, inputs, and outputs of a digital circuit design. VHDL is similar in style and syntax to modern programming languages but includes many hardware-specific constructs. In the early 1990s, VHDL was being used primarily for complex application specific integrated circuit (ASIC) design, using synthesis tools to automatically create and optimise the implementation. In the second half of the decade, the use of VHDL with synthesis has moved into the area of programmable logic design. There is also an increase in the use of VHDL for modelling specifications, both of the hardware part of the system, and of the complete system itself. It is in recent years that this language has become increasingly popular for the design of gate array devices, as the density and complexity of complex programmable logic devices (CPLDs) and field programmable gate arrays (FPGAs) based designs have increased dramatically. Several factors have pushed the development and widespread use of the hardware description language in developing electronic systems:

- The complexity of electronic systems has and is increasing much faster than the capabilities of the processes to develop the systems.
- Economic and competitive pressures are forcing development times to decrease, even though systems are becoming more complex.
- In the commercial world, time-to-market is often more important than development cost.
- The quality and reliability of electronic systems are required to be higher than ever before.
- Design teams and expertise are distributed geographically and there is need of data transfer.

Consequently with the conflicting challenges of control system complexity and the need for short, efficient design cycles, the use of high-level design languages is rapidly extending. VHDL language and VHDL development tools provide an efficient method of developing test information, a method of incorporating this information into simulations and modelling activities, a technique for evaluating simulation results, validating test plans and procedures prior to real hardware and software integration, can save significant time, effort, and rework to the design.

Using VHDL language for the design of larger CPLD/FPGA devices has shown advantages, for example

- EDA tool platform independent design (ASCII files)
- Design flexibility
- Allows design reuse
- Fully automatic design process
- Rapid prototyping in conjunction with CPLD/FPGA
- Compact construction/implementation
- Cost effectiveness
- Short time to market

and the following limitations:

1. VHDL is primarily a digital design language. It currently has very limited capabilities in the analogue area. Considerable work is going on to standardize an analogue version of the language.
2. The current standard defines a language and its syntax, without describing any styles of using it on a design project. There are other standards which exist, or are required, to define a consistent style of using the language in certain areas.
3. Currently, it is only possible to synthesize logic from a subset of the VHDL language. Each tool defines its own slightly different subset. It is possible that the code may need to be slightly modified before it can be used with a different synthesis tool set than it was originally written for.

With modern Electronic Design Automation (EDA) techniques the development of high performance Application Specific Integrated Circuits (ASICs) for electronic

systems may be rapidly developed. The implementation into Complex Programmable Logic Devices (CPLDs) and Field Programmable Gate Arrays (FPGAs) provides further advantages for industrial applications. The advanced techniques described form the basis of the implementation strategy and are used in this research program, which aims to realise a novel intelligent control system for a stand-alone generator set.

Chapter 3

The Diesel Engine

This chapter discusses the technical aspects and modelling of the diesel engine system. Firstly, the technical features of diesel engines are described, then the simulation model is developed and simulation results are obtained using the model presented.

3.1. Technical features of diesel engines

3.1.1. Diesel engine classifications and technical terms

Diesel engines may be categorised according to several different criteria, such as type of fuel, engine speed, form of aspiration and operating cycle.

Classifications

Type of fuel

Diesel engines can be fuelled by oil products ranging from No 2 diesel oil (light) to crude No 6 residual oil (heavy). The choice of fuel depends upon cost, availability, calorific value, temperature conditions and specific engine features. Smaller diesel engines generally utilise diesel oil although some are designed to be multi-fuelled. Fuel generally accounts for about 80% of the diesel generator operating costs.

Aspiration

Diesel engines can be either naturally aspirated or turbo charged. In naturally aspirated engines, air is taken in at atmospheric pressure, while turbo charged engines inject the air/fuel mixture into the engine cylinders at pressures higher than atmospheric. Turbo charging significantly increases engine capacity (up to 40%) and efficiency by allowing an increased quantity of air to enter the combustion chamber, so ensuring a

greater and more efficient burning of the fuel. Smaller diesel engines are normally naturally aspirated.

Operating cycle

Engines are classified either as two-stroke or four-stroke. In a four-stroke engine the induction, compression, power (expansion) and exhaust phases of the combustion cycle all occur on separate strokes of the piston, whereas in the two-stroke engine the exhaust and induction phases occur when the piston is almost stationary at the end of its expansion stroke. Two-stroke engines tend to be either very large or very small.

Both petrol and diesel engines are internal combustion (IC) engines working on either the two-stroke or four-stroke cycle. The basic design of both engines is similar, the main differences between the two types of engine are the method of introducing the fuel charge into the combustion chamber, and the means used to ignite it. In a diesel engine, only air is compressed on the compression stroke. The fuel charge required to produce the power stroke in each cylinder is accurately metered and pressurised by the fuel injection pump and then passes to the high pressure injectors via pipes. Once the fuel is sprayed into the combustion chamber, it is mixed with hot compressed air and ignited.

Method of cooling

The engine cooling can be air, forced air, or liquid.

Engine speed

The speed of diesel engines is generally related to size. Large engines tend to operate at lower speeds, usually less than 900 r/min. Higher speed engines (over 900 r/min) are more common in the small to medium size range. Common operating speeds are 1200, 1500, 1800, 3000 and 3600 r/min which are suited to the common grid operating frequency of 50 or 60 Hz. Engine speed is generally a trade off between size (capacity), cost, and engine life.

Technical terms [3]

Compression Ratio

Compression Ratio is the ratio of the volume of air in the cylinder when the piston is at the bottom dead centre (BDC), to the volume of air presented in the cylinder when the piston is at the top dead centre (TDC). The volume of air in the cylinder when the piston is at BDC is equal to the swept volume plus the clearance volume, and the air volume when the piston is at TDC is equal to the clearance volume, therefore

$$\text{Compression ratio} = \frac{\text{Swept volume} + \text{clearance volume}}{\text{clearance volume}}.$$

Diesel engines rely on the heat of the compressed air to ignite the fuel as quickly as possible when it is sprayed into the engine cylinders, so high compression ratios are essential.

Compression ignition

When the fuel charge is injected into the combustion chamber and mixed with the highly compressed air at a predetermined point in the cycle, the heat in the air is sufficient to cause the fuel to ignite. This is called compression ignition.

Brake thermal efficiency

It is usual to use the general term thermal efficiency to mean brake thermal efficiency. During the conversion of the chemical energy of the fuel to heat energy in the engine, approximately 30% energy is lost to the cooling system, 30% to the exhaust, 10% is used in overcoming the internal friction of the engine, leaving about 30% available to do useful work. This latter percentage is the (brake) thermal efficiency.

Mechanical fitness

In a diesel engine there is more than adequate air for combustion, therefore combustion (the burning of the fuel charge) is very nearly complete. Due to the high compression ratios and combustion process employed, a diesel engine has much higher cylinder pressure than a petrol engine of similar power. This means that the internal stresses in the diesel engine components are greater than those in a petrol

engine. Consequently the crankshaft, bearings, pistons, connecting rods and so on must be made stronger to withstand these stresses. Also, at the same speed, the inertia force acting in a diesel engine is greater than that in a petrol engine.

Compression pressure

Compression pressure may be defined as the maximum air pressure created in the cylinder of an IC engine on the compression stroke. The actual compression pressure of the air in a cylinder of an engine is related to a number of factors such as engine speed, cylinder ring wear and valve condition but the theoretical compression pressure depends on three factors:

- The pressure of the air in the cylinder before it is compressed
- The compression ratio used, and
- The gas constant for the pressure-volume relationship of the air under the conditions existing in the engine (as pressure increases, so the volume decreases, the mathematical relationship of the two for a particular gas is known as the gas constant).

The theoretical compression pressure of an engine is found by using the formula [3]

$$P_2 = P_1 r^n$$

where

P_1= pressure of the air in the cylinder before compression

P_2= theoretical compression pressure

r = compression ratio

n= the characteristic gas constant.

Compression temperature

As a mass of air is compressed, the speed of the molecules in the air is increased, heat is generated and so the temperature of the mass of air rises. Compression temperature may be defined as the temperature attained by the air, which is compressed in the cylinder of an IC engine, on the compression stroke. It should be noted that compression temperature is the temperature attained by the air due to compression in the cylinder and is not the temperature of the gases in the cylinder when combustion is in progress.

The actual compression temperature of the air in the cylinder of an IC engine is mainly dependent on the compression pressure, which is, in turn, dependent on the compression ratio and the mechanical fitness of the engine. An increase in engine rpm, however, will increase the compression pressure and subsequent compression temperature. The theoretical compression temperature of the air in the cylinder of an IC engine depends on three factors:

- The temperature of the air entering the cylinder before it is compressed
- The compression ratio used and
- The value of the characteristic gas constant.

The theoretical compression temperature of an engine may be expressed as [3]

$$T_2 = T_1 r^{n-1}$$

where

T_1= temperature of the air in the cylinder before compression begins (absolute temperature)

T_2=theoretical compression temperature (absolute temperature)

r= compression ratio

n= the characteristic gas constant.

All reciprocating piston IC engines work on either a four-stroke or two-stroke cycle. These cycles designate, in correct sequence, the mechanical actions by which (a) the fuel and air gain access to the engine cylinder (b) the gas pressure (due to combustion) is converted into power, and (c) the burnt gas is expelled from the engine cylinder.

A four stroke diesel engine is considered to be used in this project, therefore only this type of engine is described in detail. From the name, it is fairly obvious that there are four strokes in one complete engine cycle. A stroke is the movement of the piston through the full length of the cylinder, since one such movement causes the crankshaft to rotate half a turn, there are two crankshaft revolutions in one complete engine cycle. The four strokes, in correct order, are as follows:

1. *The inlet stroke* With the inlet valve opens and the exhaust valve closes, the piston moves from TDC to BDC, creating a low pressure area in the cylinder. Clean filtered air rushes through the open inlet valve to the low pressure area, and the cylinder fills with air.

2. *The compression stroke* With both valves closed, the piston moves form BDC to TDC, compressing the air. During this stroke the air becomes heated to a temperature sufficiently high to ignite the fuel.

3. *The power stroke* At approximately TDC, the fuel is injected, (sprayed into the hot and compressed air), and ignites, burns and expands. Both valves remain closed and the pressure produced acts on the piston crown forcing it down the cylinder from TDC to BDC.

4. *The exhaust stroke* At approximately BDC the exhaust valve opens and the piston starts to move from BDC to TDC, driving the burnt gas out off the cylinder through the open exhaust valve.

At the completion of the exhaust stroke, the exhaust valve closes, the inlet valve opens and the piston moves to the next inlet stroke. Since there are three non-working strokes and one working stroke, some means of keeping the engine turning over must be provided, particularly in single cylinder engines. It is for this reason that sometimes a heavy flywheel is fitted to the crankshaft to ensure smooth running.

The main characteristics of high speed diesel engines are summarised in Table 3.1 [3].

Table 3.1 Main characteristics of a high speed diesel engine

Admission of fuel	Directly from fuel injector
Compression ratio	From 14:1 to 24:1
Ignition	Heat due to compression
Torque	Varies little throughout the speed range
Brake thermal efficiency	35-40%
Engine construction	Robust
Maximum crankshaft rpm	From 2500 to 5000 rpm
Compression pressure	Actual, 3100-3800 kPa Theoretical at 16:1 CR, 4254 kPa
Compression temperature	Actual, 425-550°C Theoretical at 16:1 CR, 525°C

3.1.2. Energy conversion

Generally, from prime mover to electrical energy consumer, the power system involves a complicated energy converting and energy transferring procedure. For the diesel engine and generator system, the process is illustrated in Figure 3.1.

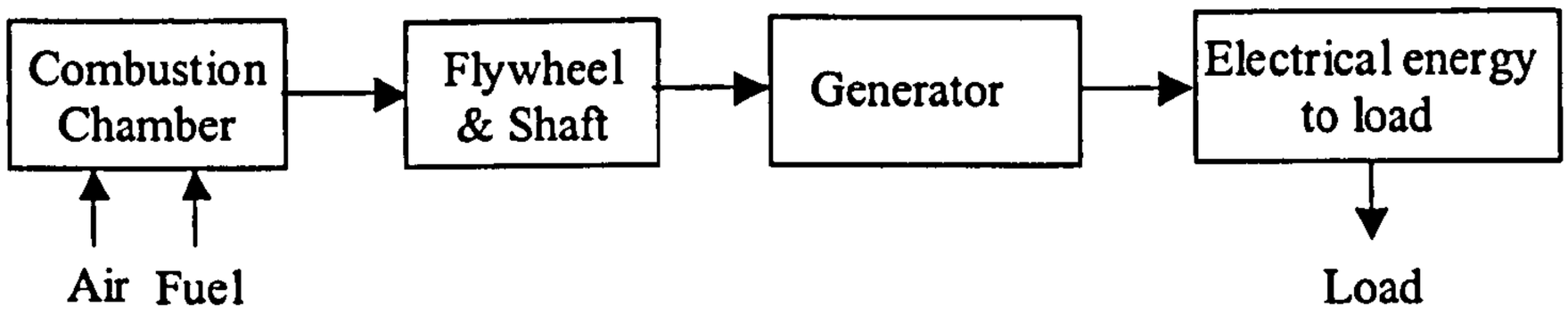


Figure 3.1. Energy converting and transferring in a diesel generator system

In a stand-alone power system, the amount of electricity generated depends on the energy demanded by the consumer. As the load increases or decreases, the power output, and so the amount of fuel input, has to be adjusted in order to satisfy consumer demand and maintain the system operating in a stable condition. Many manufacturers recommend that a diesel engine should not operate below a certain load for extended periods of time, 40% being a common threshold, although 20-30% might be more

common for some new sets. One reason is that even when idling a diesel engine will consume appreciable amounts of fuel, say 20-30% of that at full load.

3.1.3. Main factors affecting the energy conversion in a diesel engine system

In a diesel engine system, apart from mechanical construction, the main factors affecting the conversion process are [3]:

- **Admission of fuel,** which is the fuel directly injected by the fuel injector. The fuel is the source of energy conversion. Ideally the quantity or rate of fuel input should be able to respond to the output energy required by the consumer immediately and exactly, so as to provide quality electric energy to the consumer. To achieve this it is necessary to exert torque control and maintain the engine speed at the required level.
- **Air inlet,** which is essential for fuel burning in the combustion chamber. In a diesel engine, the air is compressed and mixed with fuel to experience the thermodynamic reaction, so that the kinetic energy, and then the mechanical energy is produced and delivered by the crankshaft.
- **Air temperature and pressure,** both of them are important factors in the thermodynamic procedure. In diesel engines, the highly compressed air in cylinder with high-speed movement of the molecules causes the temperature rises in the cylinder. When the fuel is introduced into the combustion chamber, it is broken up into very small droplets and spread through the compressed air. As soon as the droplets make contact with the hot air, they absorb heat from the air and burn. The greater the difference between the air temperature and the self-ignition temperature, the more rapidly the fuel vaporises and ignites. Moreover, the greater the pressure in the combustion chamber, the closer the contact between fuel droplets and hot air molecules, and therefore the greater the heat transfer rate is. This leads to a more rapid vaporisation and ignition. The compression temperature of the air in the cylinder mainly depends on the compression pressure, which is, in turn, dependant on the compression ratio and the mechanical fitness of the engine. The engine speed can be another factor if a turbocharger is used. The engine speed increase or decrease will

cause the compression pressure and subsequent compression temperature to increase or decrease.

- **Altitude.** When the diesel engine system is operated at high altitude where the air density is less than that at sea level, the quantity of air and oxygen entering the engine cylinder on inlet stroke may be insufficient for the combustion of the normal fuel charge. Also as altitude increases, air density decreases, this causes the engine power to decrease, and will seriously affect engine performance in proportion to the altitude. Power decreases approximately 3 percent of rated power each 1000ft (305m) of increase in elevation above sea level. Table 3.2 shows the relation between the maximum output power and altitude for a typical 8kW Genset [36].

Table 3.2 Power vs altitude—8kW GENSET

Elevation above Sea Level	Maximum Genset Power
at/below 500ft (152m)	8000W (rated)
at 2500ft (762m)	7520W
At 5500ft (1676m)	6800W
above 5500ft (1676m)	subtract 240W each additional 1000ft (305m)

- **The fitness of the fuel particles and injection pressure.** There are two factors affecting fuel combustion. If the fuel is broken up into fine enough particles (atomised), the vaporisation required for combustion would be negligible and ignition would start almost immediately. However, the mass of the fuel particles under these conditions would not be sufficient to carry the particles far from the injector nozzle, and complete combustion would not occur. For complete combustion, good depth of penetration of the fuel particles into the combustion chamber is necessary, and the particles must have sufficient mass to carry them deep into the compressed air. Again, if the particles are very fine, the total surface area presented to the compressed air will be large and a great volume of fuel will be vaporised almost immediately. Once combustion begins, all the vaporised fuel will be burnt rapidly, and a rapid and high-pressure rise will occur in the combustion chamber.

Other factors affecting the energy conversion are the purity of the fuel, the temperature of the air entering the cylinder before it is compressed, the value of the characteristic gas constant, the air being used (or the combination of the air and its characteristic constant), and the condition of the air (dust, dirt particulate).

Some of the factors can be mechanically improved, for instance, using an air-cleaner, fuel-filters, turbocharger and intercoolers. Obviously an air-cleaning system is responsible for removing dust from the air and the fuel-filtering system is responsible for purifying the fuel, while a turbocharger increases the air volume admitted to the cylinder. The turbocharger is essentially an exhaust-driven supercharger, its primary purpose being to pressurise the intake air from the air-cleaner. Intercoolers are used to reduce the high temperature of the air leaving the turbocharger, so as to increase the density and quantity of the intake air entering the engine cylinder on the inlet stroke. As air pressure increases the temperature increases, therefore allowing more fuel to burn effectively, and consequently developing a greater power and torque output from the engine. The system must be real-time adjusted to reflect power output and torque conditions when the system load changes.

3.1.4. Diesel engine governor and speed control

To operate a diesel engine generator system efficiently, the speed of the engine and generator has to be varied with changing load. To realise this, a speed governing system plays an important role, controlling the fuel input of the engine and hence the engine and generator speed. The diesel engine governing system controls the engine speed by adjusting fuel flow, so as to control the generator output and provide sufficient power to match demand.

Any device that automatically exerts control over engine speed may be termed a governor. It is obvious that a governor is required on a diesel engine to:

- prevent stalling and over speeding and/or
- maintain engine speed at the desired level regardless of load variations.

Governors are generally classified by the function they fulfil, for example

- constant speed governors

- variable or all speed governors
- idling and maximum speed governors.

Apart from their governing characteristics, governors take many forms and are actuated in many different ways, for example mechanically (or centrifugal), pneumatically, hydraulically or electronically. For example, sensitive hydraulic governors exercise delicate control over the quantity of fuel injected, increasing or decreasing it as required to maintain the engine speed at their required level. Electronic governors deal with input variables such as ambient temperature, turbocharger boost, fuel temperature and many other factors, to effect precise speed control and low emission operation. Electronic governing system can be used in processing control, which could be designed in open loop or closed loop.

In general a mechanical governor controls the prime mover during start up and then acts as a backup governor. The overall block diagram of a diesel engine generation system is shown in Figure 3.2, where the engine drives a stand-alone generator to operate at fixed output frequency and voltage amplitude under varying load conditions. In addition, some manufactures make governors that operate on a combination of two of the above principles, for example, mechanical-hydraulic and pneumatic-mechanical governors.

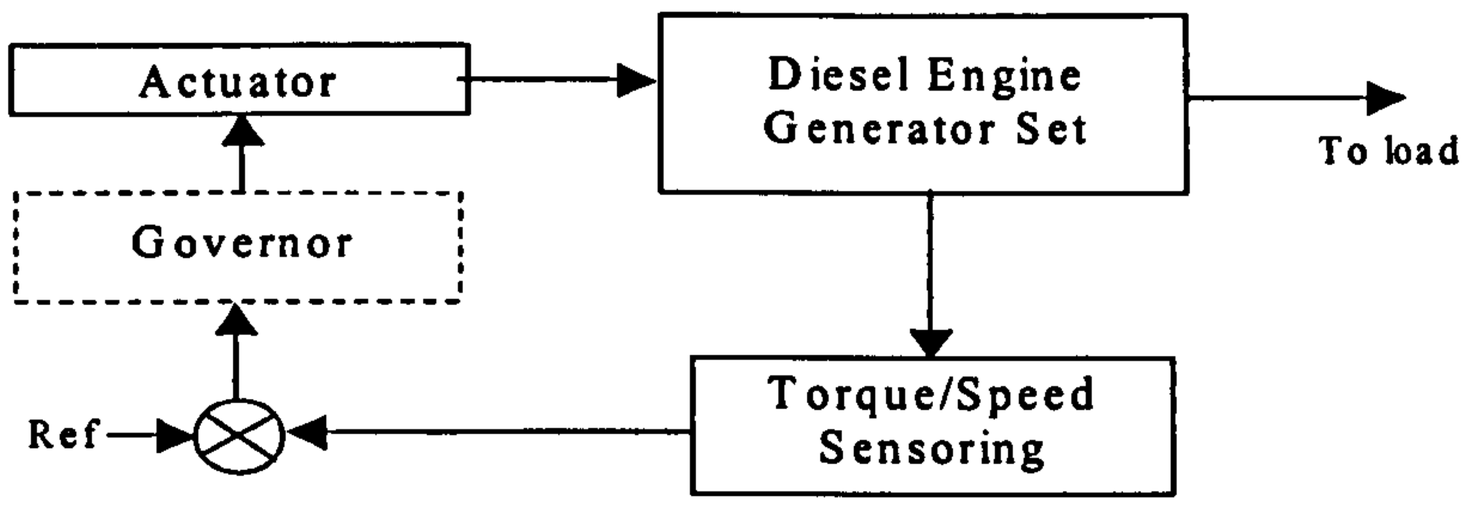


Figure 3.2. Block diagram of a conventional speed control system

In Figure 3.2 the dashed line framed block is the conventional governor system, which operates on the basis of the shaft speed. In contrast, the proposed control system shown in grey in Figure 3.3 is based on the electrical variables that reflect the changing load and speed. The detailed control strategy and the control system design are discussed later in this thesis.

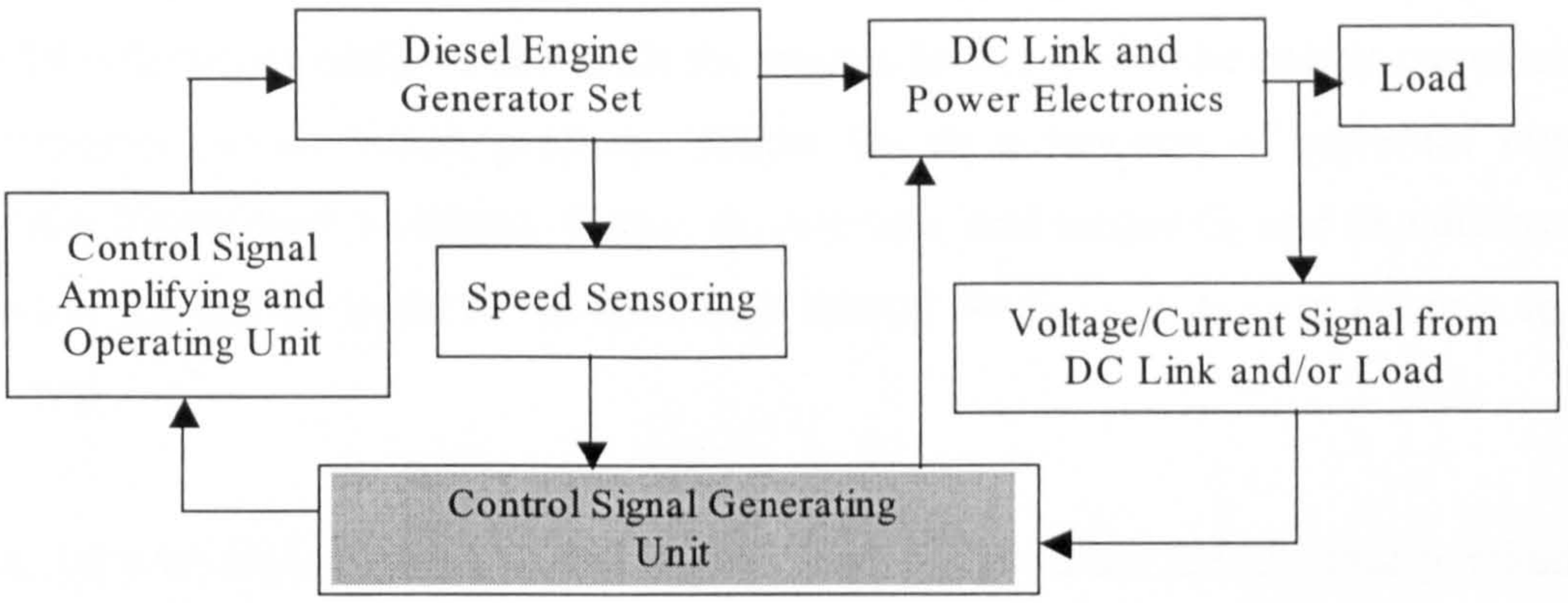


Figure 3.3. Block diagram of the proposed control system

3.2. Diesel engine system modelling

Before commencing on a detailed description of control strategy and controller design, the fundamental characteristics of the diesel engine viewed as an element in a control system are discussed.

The diesel engine is a very complex device and there are many non-linear factors that affect diesel engine performance, consequently it is difficult to formulate a comprehensive mathematical model. Fortunately a really detailed mathematical model is not necessary for a basic understanding of control action. However unless a transient analysis is required, a simple 'small incremental signal' model can be used for the study of normal operating conditions. Model parameters based on empirical methods are usually sufficient.

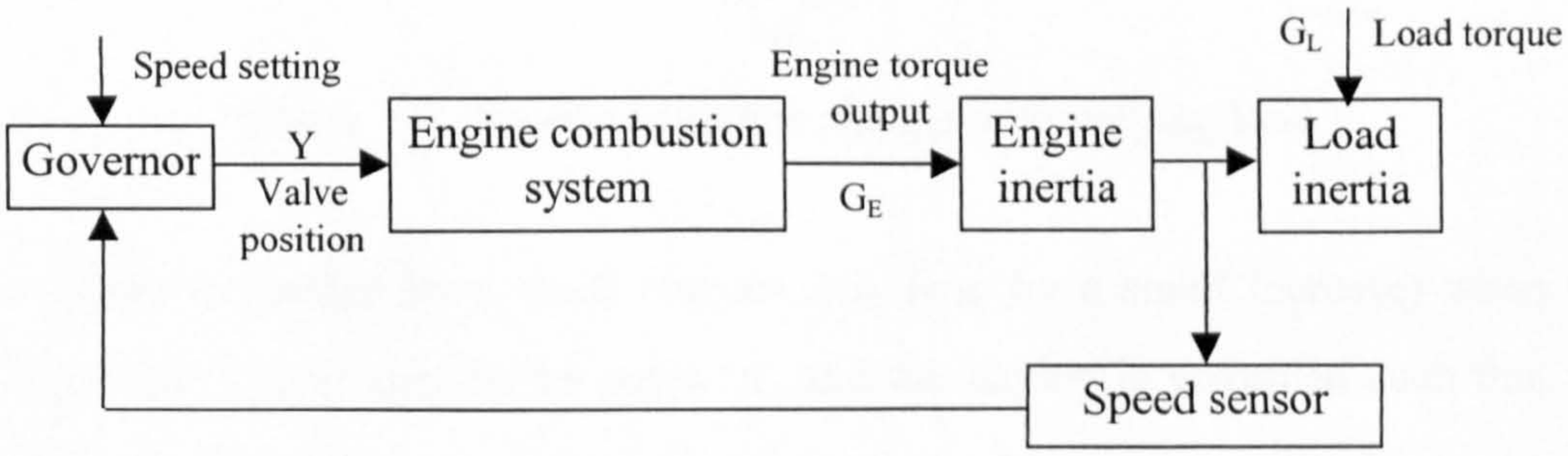


Figure 3.4. Basic engine speed control loop

The block diagram in Figure 3.4 shows a basic engine/governor control loop with a solidly coupled generator which loads the engine. In this model the engine comprises a combustion system which produces torque G_E as a function of governor output position Y (fuel rack position). Torque G_E provides load torque G_L and any difference between G_E and G_L to drive the combined inertial $J = J_{Engine} + J_{Load}$ to cause a speed variation.

It can be seen from Figure 3.4. that engine, load and governor are all closely linked to make up a complete control loop and the characteristics of all parts are equally important in determining a good governing performance.

3.2.1. Diesel engine mathematical model

Considering the normal engine and load torque/speed characteristics shown in Figure 3.5.

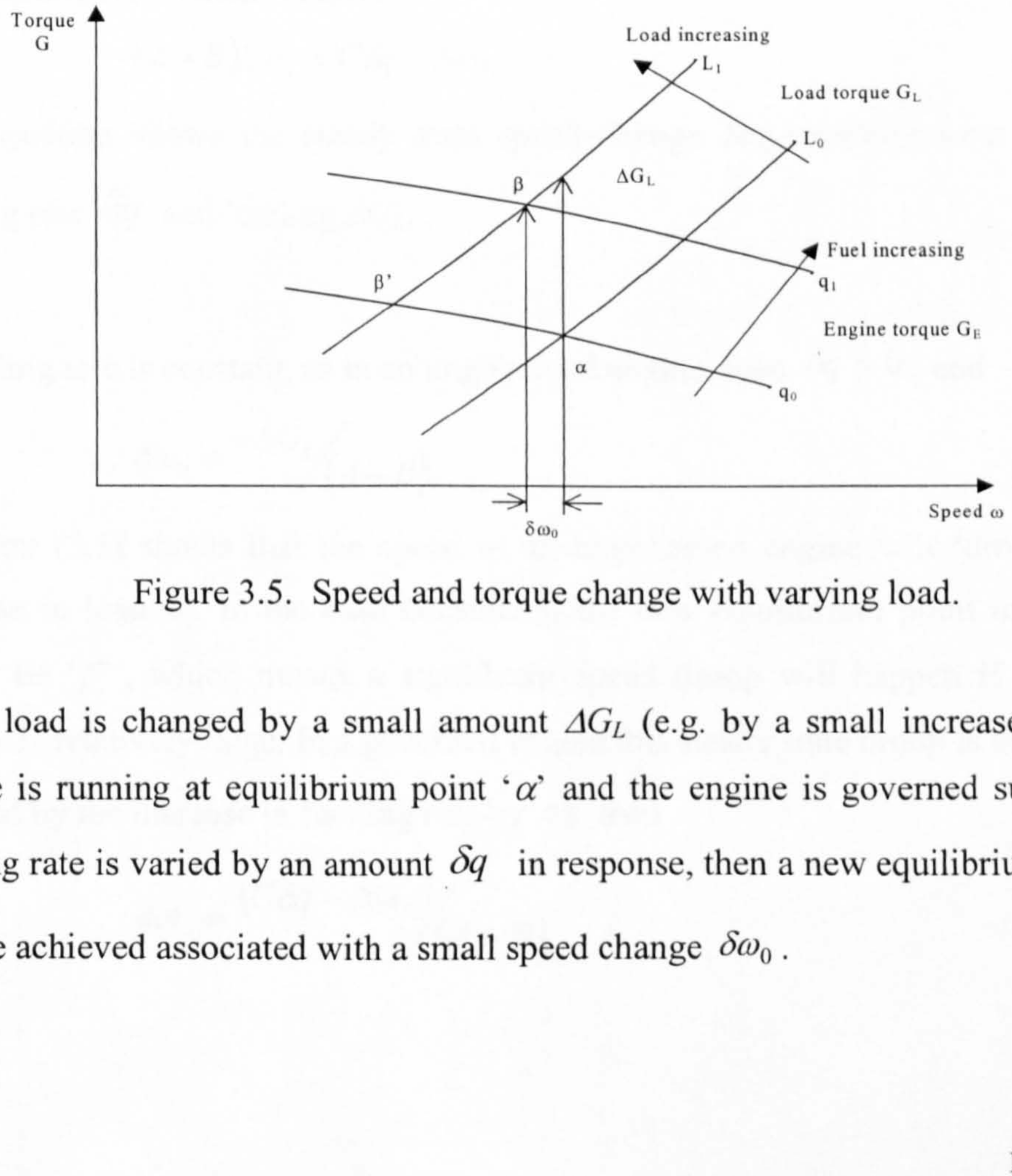


Figure 3.5. Speed and torque change with varying load.

If the load is changed by a small amount ΔG_L (e.g. by a small increase) when the engine is running at equilibrium point 'α' and the engine is governed such that the fuelling rate is varied by an amount δq in response, then a new equilibrium point 'β' will be achieved associated with a small speed change $\delta\omega_0$.

The final change in the driving torque is given by:

$$\delta G_E = \left(\delta G_E / \delta \omega_0 \right)_q \delta \omega_0 + \left(\delta G_E / \delta q \right)_{\omega_0} \delta q$$

The variation of load torque after taking speed variation into account is:

$$\delta G_L = \Delta G_L + \left(\delta G_L / \delta \omega_0 \right)_L \delta \omega_0$$

For equilibrium

$$\delta G_E = \delta G_L ,$$

then

$$\left(\delta G_E / \delta \omega_0 \right)_q \delta \omega_0 + \left(\delta G_E / \delta q \right)_{\omega_0} \delta q = \Delta G_L + \left(\delta G_L / \delta \omega_0 \right)_L \delta \omega_0 \qquad (3.1)$$

If we define that

$$\left(\delta G_L / \delta \omega_0 \right)_L = A, \quad \left(\delta G_E / \delta \omega_0 \right)_q = B, \quad \left(\delta G_E / \delta q \right)_{\omega_0} = C,$$

and rearrange (3.1) then we have

$$(A - B)\delta \omega_0 = C \delta q - \Delta G_L \qquad (3.2)$$

This equation shows the steady state speed change $\delta\omega_0$ resulting from changes in fuelling rate δq and loading ΔG_L.

If fuelling rate is constant, as in an ungoverned engine, then $\delta q = 0$, and

$$\delta \omega_0 = -\Delta G_L / (A - B) \qquad (3.3)$$

Equation (3.3) shows that the speed of an ungoverned engine will 'droop' for any increase in loading. In the case considered the new equilibrium point in Figure 3.5 would be 'β'', which means a significant speed droop will happen if the loading change is relatively large. In a governed engine this steady state droop is automatically reduced by the increase in fuelling rate by δq then

$$\delta \omega_0 = (C \delta q - \Delta G_L) / (A - B) \qquad (3.4)$$

The amount of speed droop with a governor can be reduced to zero by making $C\delta q = \Delta G_L$ but this may cause stability problems. Dynamically, during the transition between equilibrium points the torque imbalance will act on the rotating inertia thus:

$$B\delta\omega_0 + C\delta q - \Delta G_L - A\delta\omega_0 = J\left(d(\delta\omega_0)/dt\right)$$

Using s notation, where $s = d/dt$, then:

$$\delta\omega_0 = \left(C\delta q - \Delta G_L\right)/\left(Js + (A - B)\right) \qquad (3.5.a)$$

Now as δq is the fuel change caused by the governor acting, then the operation can be defined by the equation $C\delta q = K\delta y$, where y is the actuator output position, K is a constant depending on the engine and governing systems. Hence,

$$\delta\omega_0 = \left(K\delta y - \Delta G_L\right)/\left(Js + (A - B)\right) \qquad (3.5.b)$$

Equation (3.5) is the simple basic transfer function for a governor controlled engine. If $A = B = 0$ and $\Delta G_L = 0$ then the engine transfer function becomes:

$$\delta\omega_0/\delta y = K/Js \qquad (3.6)$$

Equation (3.6) shows that the engine characteristic is that at a nearly pure integrator. In general, an integrator will continue to increase its output (i.e. speed in the case of an engine) as long as there is positive input (fuel, in the case of an engine). This explains the reason why a diesel engine, in which $B \cong 0$, has a tendency to run away if it is not governed [20].

3.2.2. Combustion delay

From the definitions of $\left(\delta G_E/\delta q\right)_{\omega_0} = C$ and $C\delta q = K\delta y$ then:

$$\left(\delta G_E/\delta q\right)_{\omega_0}\delta q = K\delta y$$

therefore

$$\delta G_E/\delta y = K \qquad (3.7)$$

This equation describes how the engine output torque changes as the actuator output position changes. Ideally the engine output torque is proportional to the change of governor output position. In practical terms, a very significant feature of a diesel engine, from a control point of view, is the discontinuous manner in which power is produced due to the sequential firing of a number of cylinders. This feature is important for two reasons. First, it means that there is a time delay between the action of the governor in demanding a change in fuelling rate and the subsequent response of the engine to that change. Secondly, the engine crankshaft does not rotate at a uniform speed but rather experiences a cyclic variation in torque which gives rise to a cyclic variation in speed. A sensitive governor may respond to these cyclic speed variations and exhibit a small rhythmic movement (jiggle) at its output. Such movements due to the wear induced in the governor and fuel pump mechanism are very undesirable, and normally have to be filtered out by a mechanism in the governor drive.

In a real system, the dead-time is mainly comprised of three components: (i) the time from the actuator position change until the fuel is injected into a cylinder; (ii) the time for the fuel to burn in a cylinder and produce a torque output; (iii) the time for a new torque level to be produced in a sufficient number of cylinders and to be applied to the prime mover as a whole. The response reflects the multi-cylinder nature of a diesel engine. The effective firing delay has been found empirically to be the actual time between consecutive pistons arriving at the injection point plus approximately a quarter of a revolution of the crankshaft. The effective firing delay is therefore approximately [20]

$$\tau_D = {}^{60S_e}\!/_{2Nn} + {}^{60}\!/_{4N} \text{ (second)}$$

where

S_e =2 or 4 for two-stroke or four-stroke engine

N = speed in rev/min

n = number of cylinders

This delay can be included in the engine torque and actuator output position function for the engine combustion system so that equation (3.7) becomes equation (3.8).

$$\delta G_E / \delta y = K e^{-\tau_D \cdot s} \tag{3.8}$$

This equation describes the diesel engine system more practically in so much as when the governor acts and the actuator output position changes, the engine's output torque will change but the engine output torque is proportional to the actuator output position change with some delay. Therefore the diesel engine can be represented as a non-linear subsystem in the studied stand-alone power system. Consequently a gain unit and a dead-time unit are used together to simulate the engine performance.

3.2.3. Engine simulation model

Based on the 'small incremental signal' method and equation (3.8), equation (3.9) can be written to represent a diesel engine model in a real system.

$$f(t) = K_D * v(t - \tau_D) \tag{3.9}$$

In the equation, τ_D is the dead-time representing the engine delay time, K_D is the gain, *f* and *v* are the output and input functions of the diesel engine respectively.

The Laplace transform form of equation (3.9) is

$$F(s) = K_D * e^{-s\tau_D} U(s) \tag{3.10}$$

and this may be simplified to

$$F(s) = K_D (1 - \tau_D s) U(s) \tag{3.11}$$

where *F(s)* and *U(s)* are respectively the input and output functions of the engine in Laplace form. The corresponding block diagrams are shown in Figure 3.5 and Figure 3.6.

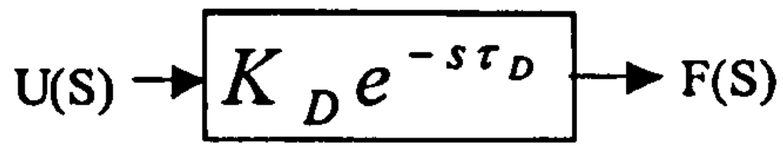


Figure 3.5. Diesel engine with dead time delay

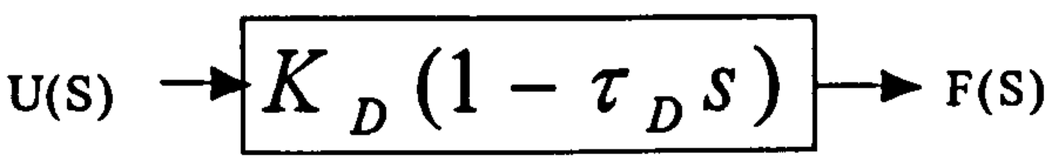


Figure 3.6. Diesel engine approximated by a simplified system transfer function

Therefore, with the delay being taken into consideration, the transfer function for a governed engine can be expressed as

$$\omega(t) = {(K_D \cdot v(t-\tau_D) - \Delta G_L)}/{(Js + (A-B))} \tag{3.12}$$

and in Laplace transform form as

$$W(s) = {(K_D \cdot e^{-\tau_D s} U(s) - \Delta G_L)}/{(Js + (A-B))} \tag{3.13}$$

from which the block diagram shown in Figure 3.7 can be constructed.

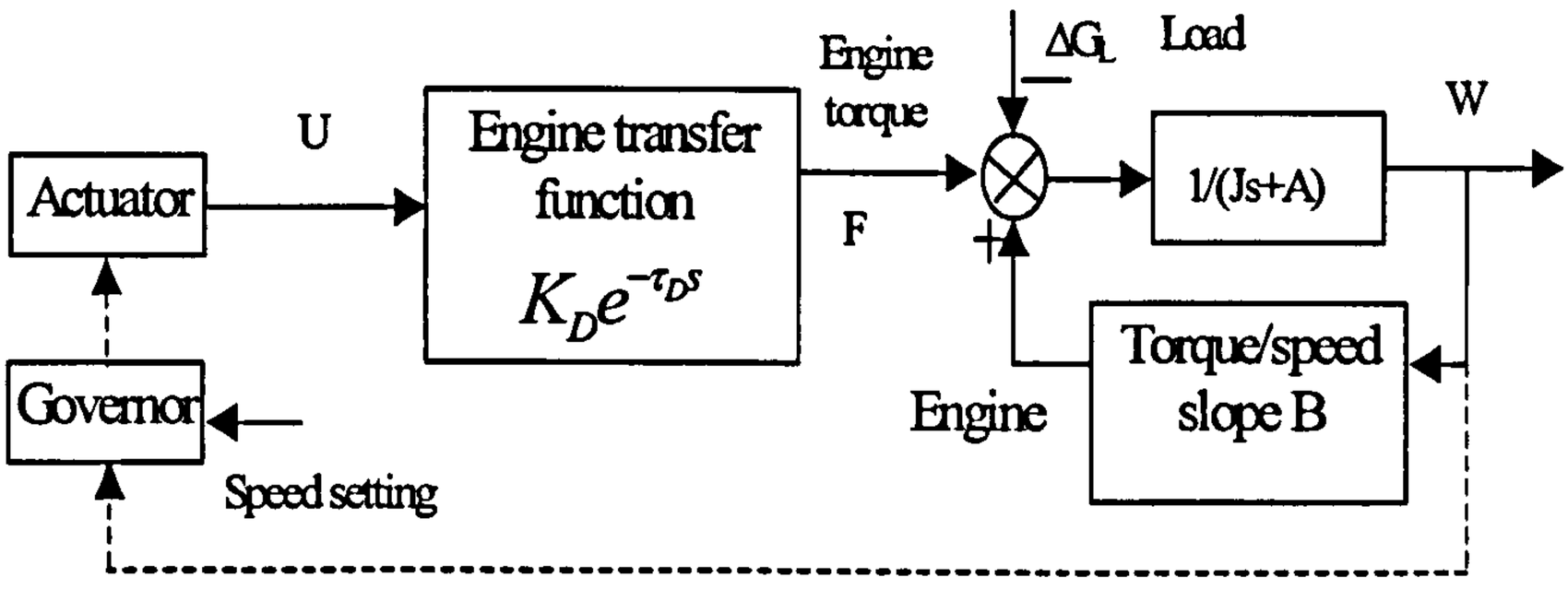


Figure 3.7. Block diagram of governed engine modelling

3.2.4. Governing system

Generally, if an engine is to be operated at a required speed, or within a required speed range, a device (governor/controller) to control the fuel in accordance with the engine load is required.

If an engine is supplying certain power at a certain speed, it will consume a corresponding amount of fuel. If the load decreases quickly and the quantity of fuel is not adjusted immediately, the difference between the energy delivered from fuel and the energy consumed by the reduced load becomes kinetic energy and cause the engine to speed up; conversely, if the load increases quickly and the fuel being injected is not able to develop sufficient power, the speed will drop. This is due to the fact that a diesel engine is fundamentally a constant torque prime mover. For any one setting of fuel control, the fuel pump delivers into each cylinder a fixed quantity of fuel at each successive injection period. Therefore, any imbalance between the fuel and the torque

required to supply the load will result in the engine either running away or stalling if appropriate control action is not taken. Hence there is need for an automatically controlled governor to adjust the amount of fuel injected into the engine cylinder. The governor responds to a change in engine speed, and adjusts the fuel injection through the fuel-control mechanism. The objectives of the control system vary with system requirements and can be either constant speed or variable speed.

In general, a control loop for an engine can be depicted by the simple flow diagram shown in Figure 3.8. Signals from the engine through the interface are sent to the control decision unit to decide an appropriate control output level, which is sent to an actuator that adjusts the fuel input to meet the system power demand.

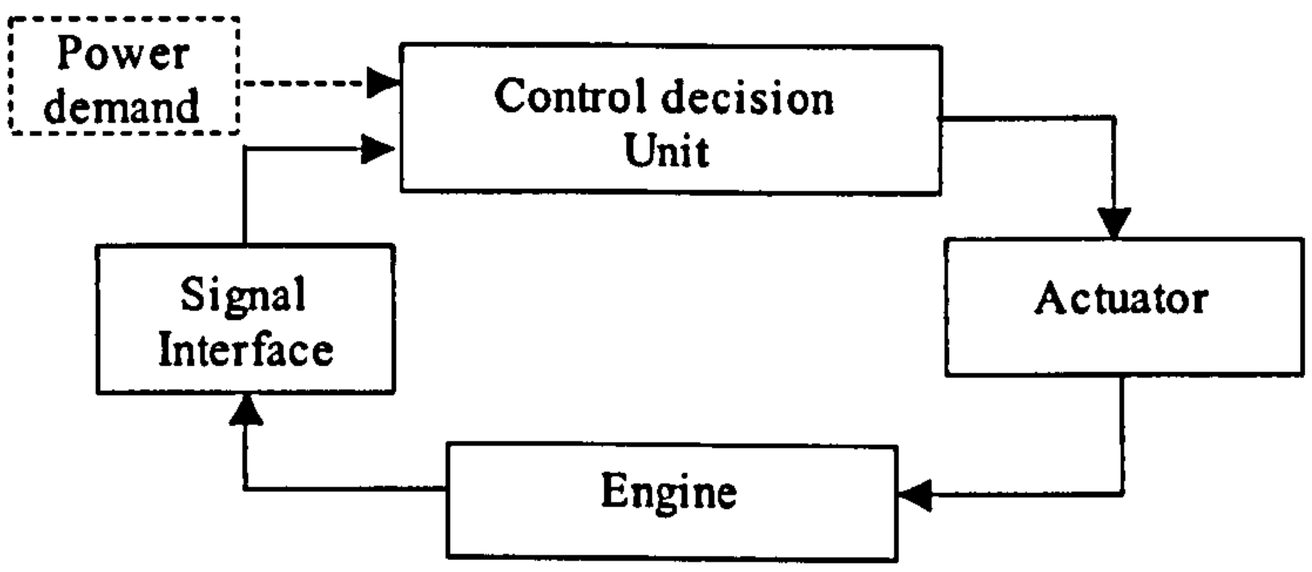


Figure 3.8. Signal flow diagram for governing system

As with other control systems there are numerous possible variations in types of electronic governor systems, however, in essence, they must all have the following components:

- Electrical power source.
- Signal sensing capability (when required).
- Electronic module for mixing, shaping and amplifying the signals.
- Control signal generation

The functions of each part in the diagram are as follows:

- The speed interface: this measures the actual engine speed. The speed sensing transducer is normally an inductive perception head generating a pulse train signal in the range 1000—5000Hz. To represent the engine

speed this signal is then integrated in a detector circuit to produce a voltage level proportional to speed.

- The control decision unit: this receives the speed signal and other necessary signals from the system, compares the actual signals with the desired setting level to produce a speed error signal '*Err*' which is then processed according to the used control method and shaped to suit different actuators. The power amplification may be quite modest if a hydraulic actuator is used with a small electro/hydraulic force motor or it may be quite substantial if an all-electric actuator based on an electric motor or large solenoid is employed.
- The actuator: this operates the fuel racks according to the control signal. The actuator receives the speed adjusting signal and electromechanically adjusts the engine's fuel supply and delivers the desired amount of fuel to the engine.
- The engine: this converts the delivered fuel into mechanical power to meet the load power and bring the engine speed to an appropriate level.

For the governor to make an appropriate control decision, accurate information on system power level requirements is very important. Normally load is derived from measurements obtained by means of current and voltage sensors. This load signal is then fed into the control decision unit.

3.2.5. Feedback control

The control function f(ε) usually consists of one or more of three elements:

a. The proportional term K_pErr. This term provides an output proportional to the error between the desired point and the actual input. With the proportional only system, the control response will be sluggish and there will be an offset in the control variable, i.e. the control will settle at a point different from that desired one. The response and percentage of the offset is directly related to the amount of the proportional gain. As the proportional gain increases, the response increases and the percentage of the offset decreases. But if the proportional gain setting is too high the system will

have the characteristic called hunting, which means the system is oscillating around the set point.

b. The rate of *Err* i.e. the derivative term $K_d d(Err)/dt$. Introducing a derivative term in the control could help to increase the system response time. In the case of a stand-alone diesel engine system, using a derivative gain can simply monitor the rate of speed change and increase the responding speed of the system. Therefore when the load on the system is changed, the derivative term anticipates the amplitude of the load change and alters actuator output, so as to adjust the fuel level before the proportional gain has time to react. Hence, the result should be a quick control of system speed while load changes.

c. The integral of *Err* as time change, i.e. the integral term $K_I \int Err \cdot dt$. The effect of the integral term is to reduce the steady state error to a minimum.

The proportional controller has no sense of time and its action is determined by the present value of the error. Using the combined actions of proportional and integral (PI) or proportional and derivative (PD) or proportional, integral and derivative (PID), more appropriate control action can be taken based on the past and predicted future values.

The controller model consists of the three terms shown in Figure 3.9, which are used in the system simulation and comparison with the controller based on the fuzzy logic method.

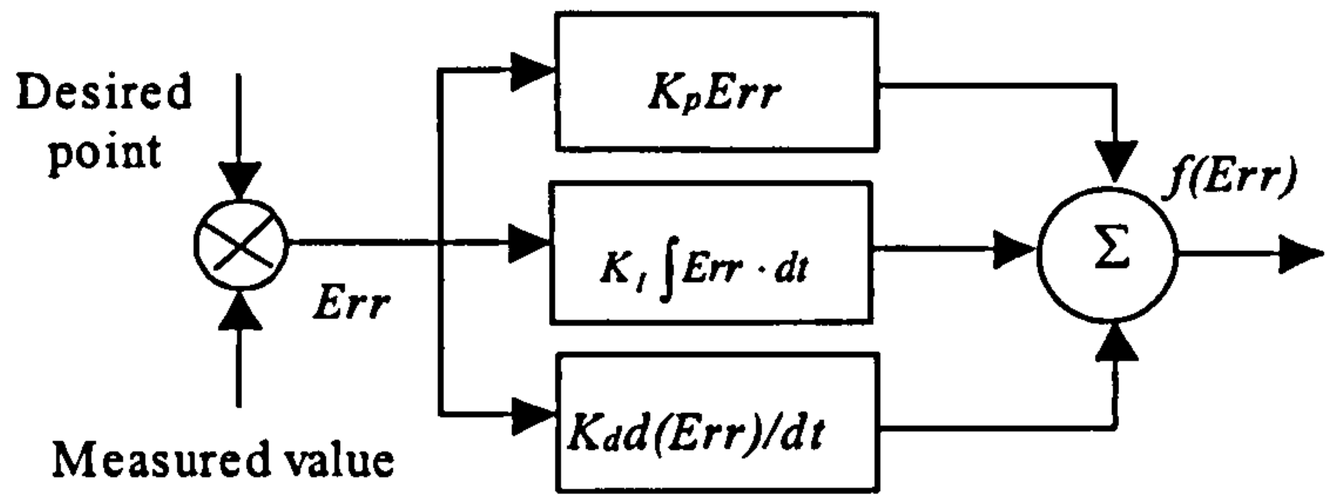


Figure 3.9. Controller model

3.2.6. Actuator

The actuator follows the controller command and executes the control action. In the system the position of the throttle lever is controlled. The prime requirements of an actuator are:

- Adequate power to effect the control
- A speed response compatible with the needs of the controller.

Actuators used in small diesel engines are mostly electromechanical, connected to a rotary throttle lever, energised by an electric solenoid. The output of the actuator is the fuel-flow, i.e. the input to the engine, U in Figure 3.10. The actuator system can be modelled by the following transfer function:

$$F(s) = \frac{K_c(1+T_1 s)}{s(1+T_2 s)(1+T_3 s)} \qquad (3.14)$$

where K_c is a gain, T_1, T_2 and T_3 are factors related to the delays in the system. If any other factors affecting the fuel flow need to be considered, a higher order transfer function could be used in modelling the actuator. However, equation (3.14) is sufficient for the simulation study requirements of this programme. A model, with a similar level of complexity, has been used for modelling a power plant with a diesel engine generator set as an auxiliary power supply in previous research [50].

Often the actuator has non-linear effects due to saturation. When developing the representation of these components, due consideration must be given to include the saturation effect as a dynamic element. To illustrate the problem, the actuator model shown in Figure 3.10 is considered. This includes a limiter block to represent the saturation effect. In the figure, A and B respectively represent the maximum and minimum limits, G is the input, x is the output response of the system if saturation is neglected and U is the output including the saturation effect.

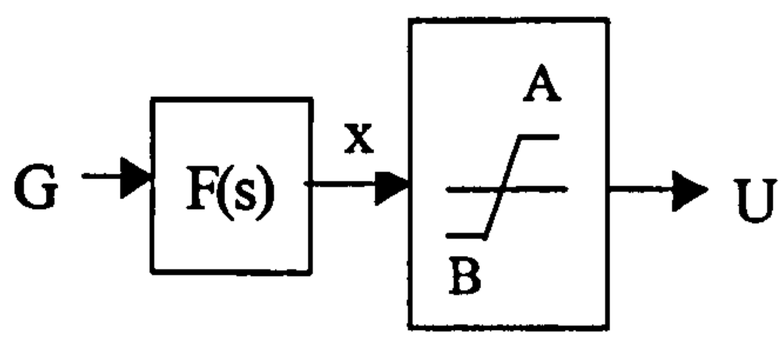


Figure 3.10. Actuator with saturation effect

3.3. Diesel engine system simulation

3.3.1. Simulation model of the system

In the foregoing the simulation models for the elements of a governed diesel engine system have been built up. With all these models connected a simulation of the system can be performed and the suitability of the models found. Simulation can allow system behaviour, with different controllers, to be studied.

For a stand-alone diesel engine generator system, in order to operate the system to meet the load demand, the power output of the engine and generator has to be varied with changing load to produce the power demanded. This is effected by appropriately rapid adjustment of the engine variables in response to the changes in power without causing unstable operating conditions.

A modelling system including the diesel engine, the actuator and the governor has been constructed by connecting together the components discussed. The model of the diesel engine system is shown in Figure 3.11, where a load is also connected to the engine system for the study of the system behaviour.

3.3.2. Simulation example of diesel engine

The simulation model of the diesel engine and control system (Figure 3.11) is built using Matlab/Simulink. The parameters used are as follows [50]:

$T_1 = 0.01$ $T_2 = 0.02$ $T_3 = 0.2$

$K = 40$ $T_D = 0.024$ $T_{max} = 1.1$ $T_{min} = 0$

K1=0.25 K2=0.009 K3=0.0384

Ke=30 kJ=10.25 K=0.125

T_{max} =0.8 T_{min} = 0

The signal to control the fuel supply is derived from the difference between the engine output power and the load. Changing load results in the change of the input signal of the controller, then the controller adjusts the input of the actuator and consequently the amount of the fuel fed to the engine. Simulation results are shown in Figures 3.12 and 3.13. Figure 3.12 shows a load curve and Figure 3.13 gives the corresponding response of the engine to the varying load. It can be seen that the engine speed varies during load changing.

The next chapter reviews the electrical generator and develops a suitable model as an element for the entire system modelling for the diesel generator model illustrated in Chapter 1 (Figure 1.1). Then the developed models will be integrated to construct the mechanical subsystem of the simulation model.

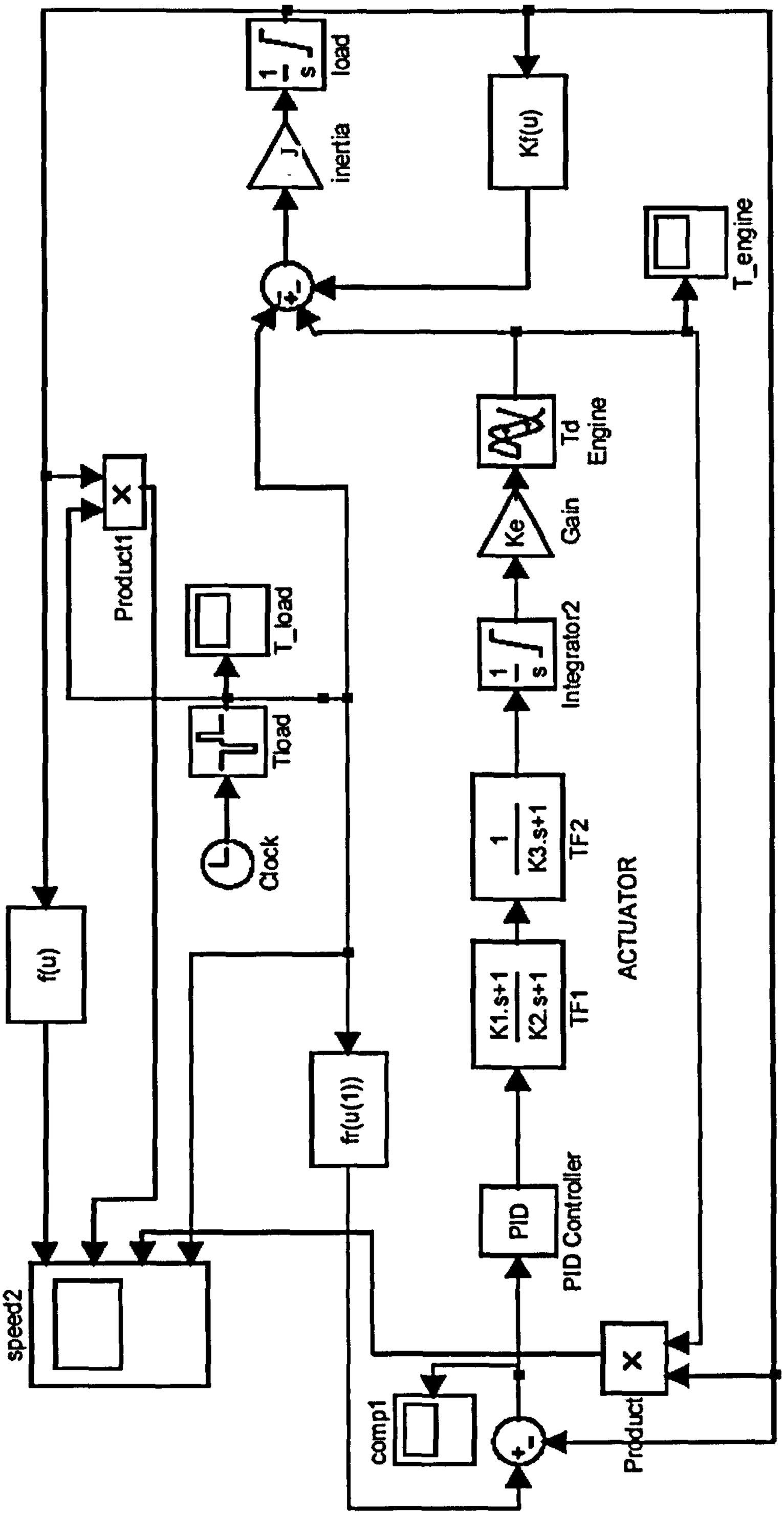


Figure 3.11. Simulation model of diesel engine system

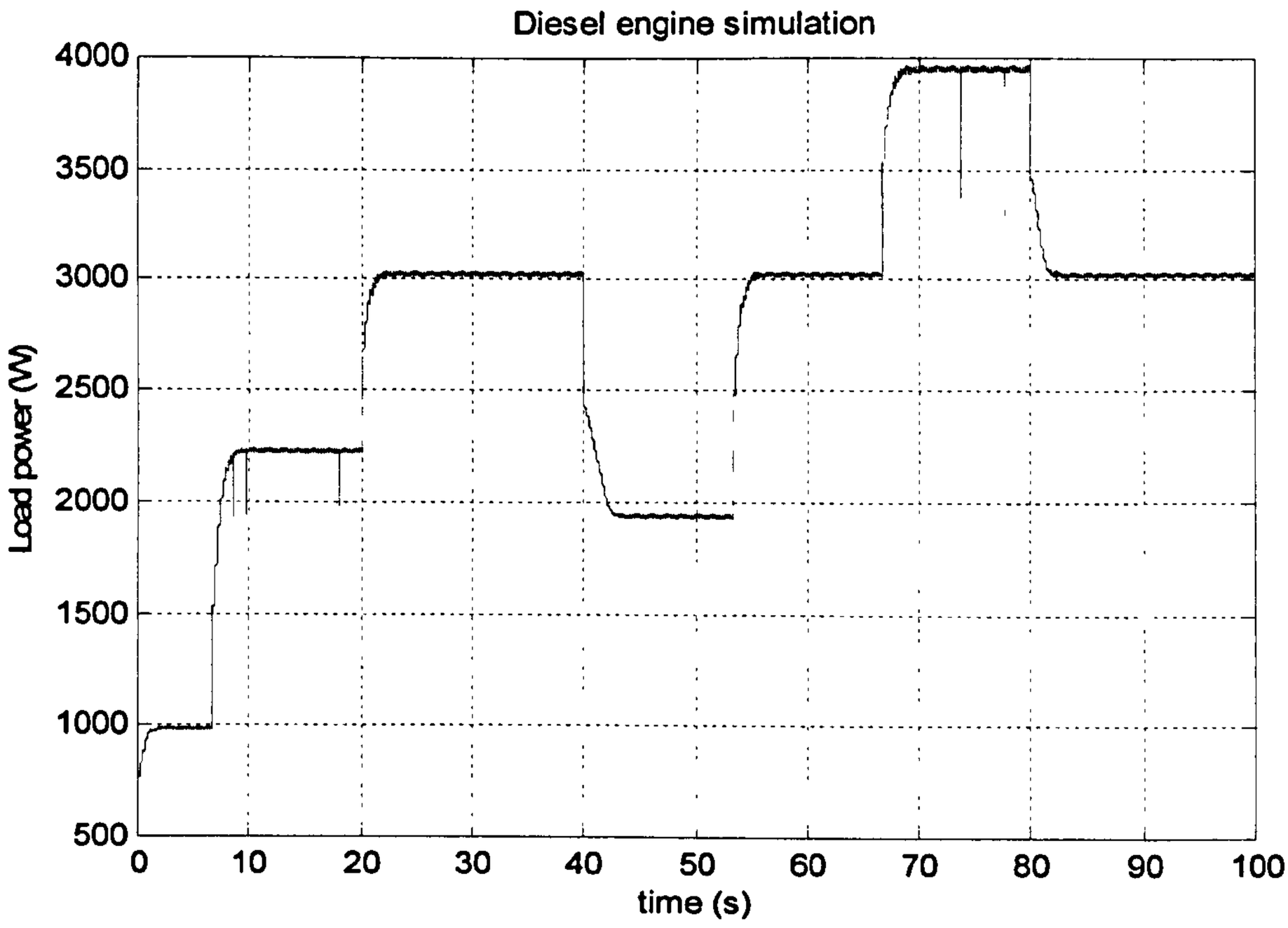


Figure 3.12. Load (W) vs time (s)

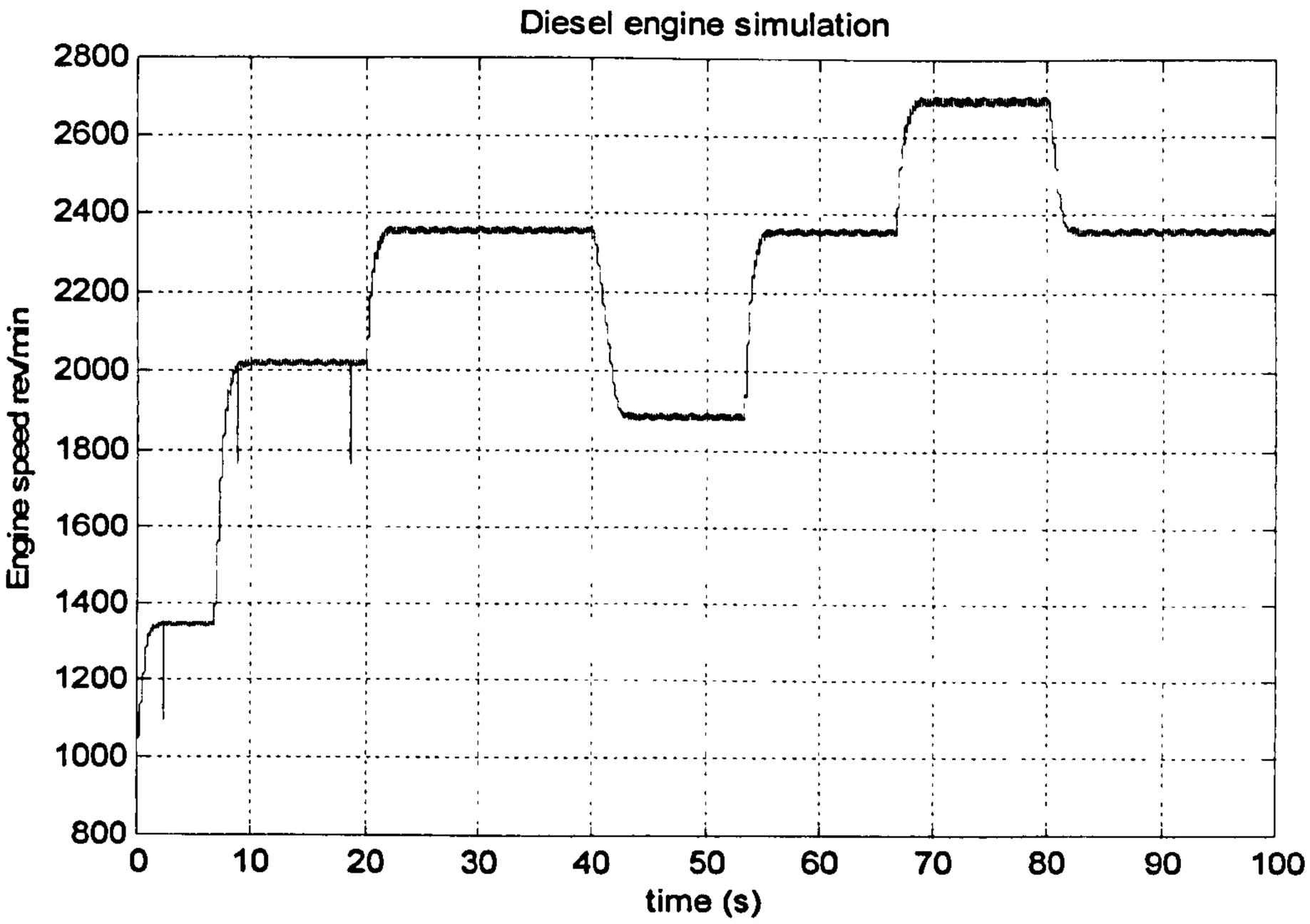


Figure 3.13. Engine speed (rev/min) vs time (s) under varying load conditions

Chapter 4

Electrical Generator

This chapter reviews electrical generators and examines the electrical and motion characteristics of generators found in typical isolated power generating systems. A mathematical model is developed and simulation work related to the generators is presented

4.1. Electrical generators

Both Permanent Magnet (PM) generators and field excited synchronous generators can be used in diesel engine driven systems. Permanent magnets used in dc and synchronous electric machines can eliminate field windings to provide loss free excitation. However, larger machines may not use permanent magnet excitation since at large scales, field windings are able to create a stronger field and produce machines of greater power/weight ratio. In large machines the cost of permanent magnet material, perceived assembly problems, and also the possibility of demagnetisation have been barriers to the use of PM machines.

The behaviour of a permanent-magnet generator is similar to that of a separately excited synchronous generator with constant excitation, therefore, the discussion of generator modelling begins with the field excited synchronous generators. The specific model for the PM machine is then derived.

Synchronous generators

The stator of a three-phase synchronous machine has a three-phase distributed winding similar to that of a three-phase induction machine. The stator winding, which is connected to an a.c. supply system, is sometimes called the armature winding. It can be designed for high voltage and current.

The rotor of the field excited generator has a winding carrying a direct current. The field winding on the rotating structure is conventionally fed from an external dc source through slip rings and brushes.

Synchronous machines can be broadly divided into two groups:

1. High-speed machines with cylindrical (or non-salient pole) rotors.
2. Low-speed machines with salient pole rotors.

Mechanical reasons such as centrifugal forces and winding assembly in specific applications have dictated the construction of these two types of rotors. The cylindrical or non-salient pole rotors are used in large generators with two or sometimes four poles and are usually driven by steam turbines. These rotors are long and have smaller diameters. On the other hand, salient pole rotors usually have large number of poles, and operate at lower speed. The rotors are shorter but have a larger diameter.

When a relatively large number of poles are necessary to produce the desired frequency, the salient-pole construction is more adequate and is employed in generators driven by relatively low speed prime movers, such as the hydraulic turbine. The salient type rotor employs concentrated windings for the field winding while the non-salient rotor uses distributed windings. This feature of the cylindrical rotor, associated with the presence of a uniform air gap, favours the production of a sinusoidally distributed mmf resulting in a practically constant armature reactance. This does not occur with the salient type and in modelling the machine, the saliency is represented by two reactances which are referred to as the direct (polar) axis reactance and the quadrature axis reactance.

Damper windings are generally fitted on the rotor of a synchronous machine. The commonest construction consists of squirrel cage bars in the pole faces connected together at the ends by rings or segments. These windings are important to stabilise the generator during transient conditions. For example when the generator is forced into

asynchronous operation (transient) the damper produces a torque that opposes the direction of the mechanical oscillations of the rotor.

The stator of the synchronous machine is made of laminated magnetic material to reduce iron losses and it is internally equipped with three identical, distributed phase windings with axes set equidistant to each other. In contrast the rotor of the synchronous machine can be solid because, in normal operation, the magnetic flux is stationary, as far as the rotor is concerned. In fact, the solid rotor iron provides current paths which act as separate damper windings during transient conditions having a beneficial effect on machine operation.

Assuming that when the field current I_f flows through the rotor field winding, a sinusoidally distributed flux is established in the air gap. Then if the rotor is now rotated by the prime mover (which can be a steam turbine or a diesel engine), a revolving field is produced in the air gap. The rotating flux produces a varying flux linkage with the armature windings and voltages are induced in the stator windings. These induced voltages in the balanced three phase windings are phase shifted by 120 electrical degrees. The rotor speed and frequency of the induced voltage are related by

$$n = \frac{120f}{p} \qquad \text{or} \qquad f = \frac{n \cdot p}{120} \tag{4.1}$$

where n is the rotor speed in rpm and p is the number of poles.

The induced rms voltage is

$$E_f = 4.44 f \cdot \Phi_f N \cdot K_w \tag{4.2}$$

where Φ_f is the flux per pole due to the excitation current I_f. N is the number of turns in each phase, and K_w is the winding factor.

It can be seen that the excitation voltage is proportional to the machine speed and excitation flux, and the latter in turn depends on the excitation current I_f. If the stator terminals of the machine are connected to a three phase load, a stator current I_a will flow. The frequency of I_a will be the same as that of the induced voltage E_f. The stator currents flowing in the three phase windings also establish a rotating field in the air

gap known as the armature reaction flux. The air gap flux is produced by both the rotor current I_f and stator current I_a.

Excitation systems

The automatic voltage regulator or, more precisely, the excitation system is composed of three major parts: the exciter, the regulator and the manual control. These components are defined by the IEEE committee report [26] as follows:

> The exciter is the source of all or part of the field current for the excitation of the synchronous machine. The regulator couples the output variables of the synchronous machine to the input of the exciter through feedback and forward controlling elements for the purpose of regulating the synchronous machine variables. The manual control consists of those elements in the excitation control system which provide for manual adjustment of the synchronous machine by open loop control.

According to the way the exciter derives its energy, the excitation system is classified into three types:

i) DC excitation system

ii) AC excitation system

iii) Static excitation system

The first two types utilise rotating machines as a power source for the excitation system, commonly a direct-current generator with commutator for the first type and an alternating-current generator with rectifier for the second. The third type, the static excitation system is typical of modern AVRs, uses transformers associated with rectifiers to provide the necessary direct current to the generator field.

4.2. Modelling and simulation of electrical generators

4.2.1. Modelling of a field excited generator

Mathematical model frame

The representation of the synchronous machine for power system analysis has been the subject of a great number of papers and articles [21][37]. Initially a simple model which represented the synchronous machine by an alternating voltage, varying in both magnitude and phase, and connected in series with an inductance and a resistance was used. However with the advent of Park's theory and later the application of digital computer, new improvements have been made in the development of a number of synchronous machine models with varying complexities.

The modelling of synchronous and induction machines for use in transient stability analysis of power systems has been based on generalised machine theory [1] (attributed principally to Park [37] and Kron [30]).

The mathematical models for synchronous machines may be represented using particular reference frames. Hancock [22] describes five of these frames:

i) The three-phase frame or a-b-c frame;

ii) The two-phase frame or α-β-0 frame;

iii) The two-axis frame or d-q-0 frame;

iv) The symmetrical component axes or p-n-0;

v) The forward and backward axes or f-b-0.

The symmetrical component axes and the forward and backward axes are more important for studies involving unbalanced conditions [22] which are not intended here. The three-phase frame uses the actual variables but the solution of the differential equations is complicated by the presence of time-varying parameters. The two-phase α-β-0 frame reduces the number of equations and maintains the time varying quantities. However, these quantities may be made time-invariant if a proper two-axis

frame is selected and this is one of the main reasons for this frame having wide application in power system analysis since the computational burden is greatly reduced during simulations. The two-axis reference frame may be either:

i) attached to the stator-stationary frame,

ii) attached to the rotor and rotating at the same speed (Parks reference frame),

iii) rotating at synchronous speed (Kron's reference frame) or

iv) rotating at an arbitrary speed chosen according to the convenience.

The choice of frame should be made to simplify the machine electrical equations, such as to obtain constant coefficients.

When performing inter-frame transformations it is considered important, though not essential, to maintain the total power invariant so that variables, such as power and torque, can be calculated directly from the transformed variables. This may be done by choosing an appropriate orthogonal transformation matrix or by using adequate factors in the power and torque equations to achieve the power-invariance. The machine model is detailed in the following sections.

Voltage equation in phase frame

The model of the synchronous machine consists of three stator windings, and one field winding on the rotor. Two additional windings could be added to the rotor, one along the direct axis and the other along the quadrature axis, to model the short-circuited paths of the damper windings.

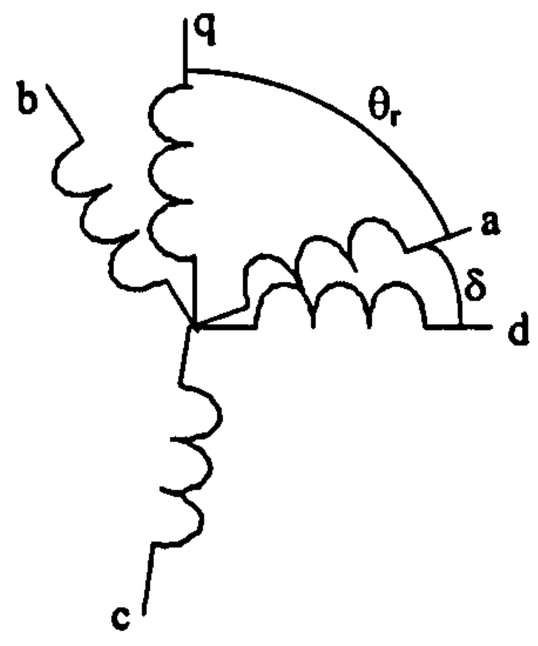


Figure 4.1 Magnetising axes of synchronous generator windings

Assuming a reference frame (axis) rotating with the synchronous speed ω, which will be coincident with the axis of phase a at t=0, θ_r is the angle by which rotor quadrature axis is ahead of the magnetic axis of phase a, and δ is the displacement of the direct axis from the synchronously rotating reference axis. The magnetising axes of the windings are shown schematically in Figure 4.1.

In the classical method, the idealised synchronous machine is represented as a group of magnetically coupled circuits with inductance, which varies with the angular position of the rotor. In addition, saturation is often neglected and spatial distribution of armature mmf is assumed sinusoidal. The circuits are schematically shown in Figure 4.2.

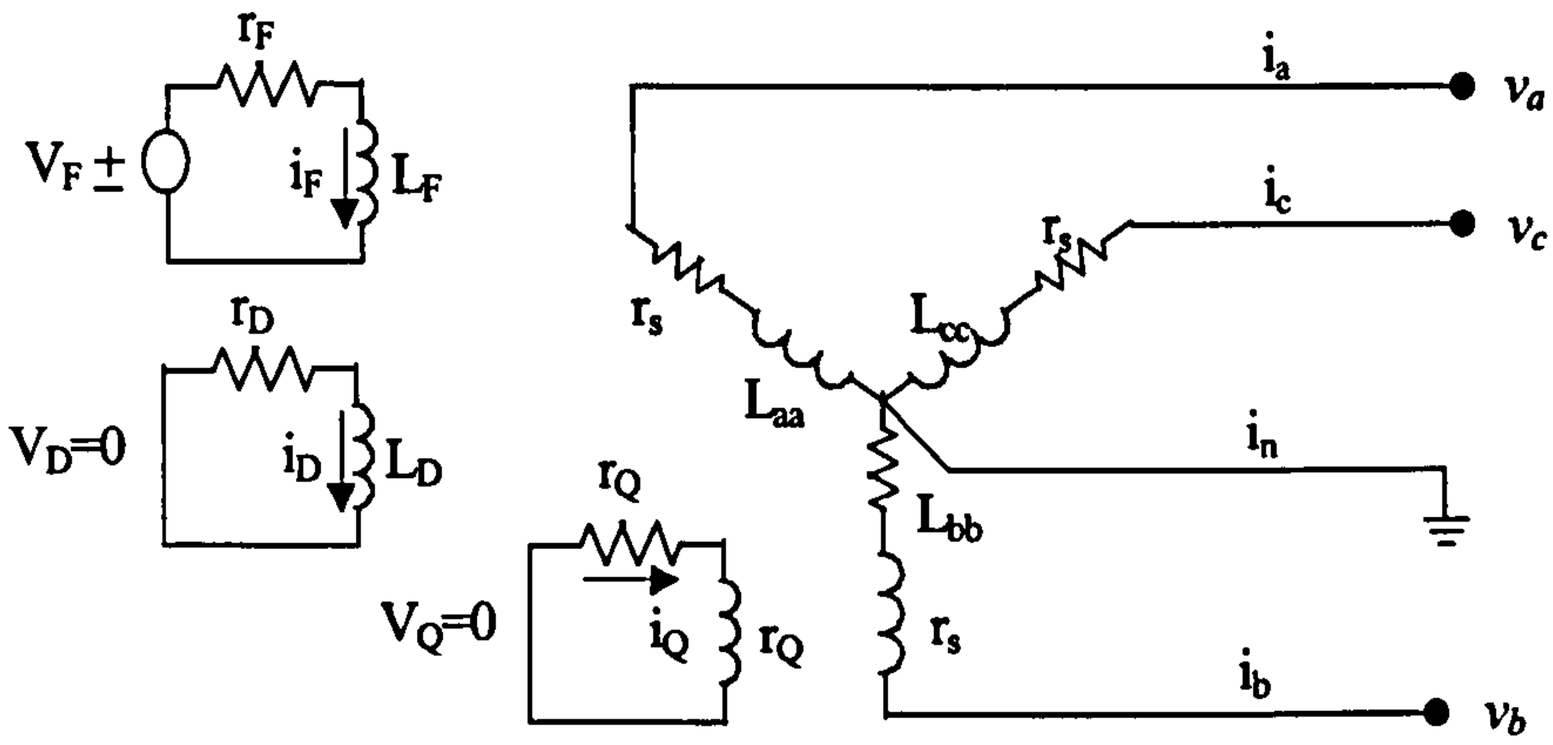


Figure 4.2 Schematic representation of mutually coupled circuits

The voltage equations of the electric machine can be expressed in the phase frame as [39]:

$$[V] = [R] \cdot [i] + \frac{d}{dt}[\lambda] \tag{4.3}$$

Where

$$[V]=\begin{bmatrix} v_a \\ v_b \\ v_c \\ v_F \\ v_D \\ v_Q \end{bmatrix} \quad [i]=\begin{bmatrix} i_a \\ i_b \\ i_c \\ i_F \\ i_D \\ i_Q \end{bmatrix} \quad [\lambda]=\begin{bmatrix} \lambda_a \\ \lambda_b \\ \lambda_c \\ \lambda_F \\ \lambda_D \\ \lambda_Q \end{bmatrix} \quad [R]=\begin{bmatrix} r_s & & & & & \\ & r_s & & & & \\ & & r_s & & & \\ & & & r_F & & \\ & & & & r_D & \\ & & & & & r_Q \end{bmatrix}$$

r_s is stator phase winding resistance and r_F, r_D, r_Q are the resistance of the field winding and direct and quadrature axis damping windings respectively.

The above equation may be written in partitioned matrix form as

$$[V_{abc}]=[R_s]\cdot[i_{abc}]+\frac{d}{dt}[\lambda_{abc}] \tag{4.4}$$

and

$$[V_{FDQ}]=[R_{FDQ}]\cdot[i_{FDQ}]+\frac{d}{dt}[\lambda_{FDQ}] \tag{4.5}$$

Where

$$[R_s]=\begin{bmatrix} r_s & 0 & 0 \\ 0 & r_s & 0 \\ 0 & 0 & r_s \end{bmatrix} \qquad [R_{FDQ}]=\begin{bmatrix} r_F & 0 & 0 \\ 0 & r_D & 0 \\ 0 & 0 & r_Q \end{bmatrix}$$

$[V_{abc}]=[V_a\ V_b\ V_c]^T$, $[i_{abc}]=[i_a\ i_b\ i_c]^T$, $[\lambda_{abc}]=[\lambda_a\ \lambda_b\ \lambda_c]^T$,

$[V_{FDQ}]=[V_F\ V_D\ V_Q]^T$, $[i_{FDQ}]=[i_F\ i_D\ i_Q]^T$ and $[\lambda_{FDQ}]=[\lambda_F\ \lambda_D\ \lambda_Q]^T$.

The flux-linkage equations are:

$$[\lambda_{abc}]=[L_{ss}]\cdot[i_{abc}]+[L_{sr}]\cdot[i_{FDQ}] \tag{4.6}$$

$$[\lambda_{FDQ}]=[L_{rs}]\cdot[i_{abc}]+[L_{rr}]\cdot[i_{FDQ}] \tag{4.7}$$

The inductance sub-matrices of the flux-linkage equations are:

$$[L_{ss}]=\begin{bmatrix} L_{aa} & L_{ab} & L_{ac} \\ L_{ba} & L_{bb} & L_{bc} \\ L_{ca} & L_{cb} & L_{cc} \end{bmatrix}$$

$$[L_{rr}]=\begin{bmatrix} L_{FF} & L_{FD} & L_{FQ} \\ L_{DF} & L_{DD} & L_{DQ} \\ L_{QF} & L_{QD} & L_{QQ} \end{bmatrix}$$

$$[L_{sr}]=[L_{rs}]^T=\begin{bmatrix} L_{aF} & L_{aD} & L_{aQ} \\ L_{bF} & L_{bD} & L_{bQ} \\ L_{cF} & L_{cD} & L_{cQ} \end{bmatrix}$$

Where the self-inductances of the stator windings are:

$$L_{aa} = L_{ls} + L_{0s} - L_{2s}\cos 2\theta_r$$

$$L_{bb} = L_{ls} + L_{0s} - L_{2s}\cos(2\theta_r + 2\pi/3)$$

$$L_{cc} = L_{ls} + L_{0s} - L_{2s}\cos(2\theta_r - 2\pi/3)$$

in which L_{ls} is stator phase leakage inductance, L_{0s} and L_{2s} are the components of magnetising inductance.

The mutual inductances of the stator windings are:

$$L_{ab} = L_{ba} = -\frac{1}{2}L_{ls} - L_{2s}\cos(2\theta_r - 2\pi/3)$$

$$L_{bc} = L_{cb} = -\frac{1}{2}L_{ls} - L_{2s}\cos 2\theta_r$$

$$L_{ca} = L_{ac} = -\frac{1}{2}L_{ls} - L_{2s}\cos(2\theta_r + 2\pi/3)$$

The self-inductances of the rotor windings are:

$$L_{FF} = L_F \qquad L_{DD} = L_D \qquad L_{QQ} = L_Q$$

The mutual inductances of the rotor windings are:

$$L_{FD} = L_{DF} = M_R \qquad L_{FQ} = L_{QF} = 0 \qquad L_{DQ} = L_{QD} = 0$$

The mutual inductances between the stator and the rotor windings are:

$$L_{aF} = L_{Fa} = L_{sF}\cos\theta_r$$

$$L_{bF} = L_{Fb} = L_{sF}\cos(\theta_r - 2\pi/3)$$

$$L_{cF} = L_{Fc} = L_{sF}\cos(\theta_r + 2\pi/3)$$

$$L_{aD} = L_{Da} = L_{sD}\cos\theta_r$$

$$L_{bD} = L_{Db} = L_{sD}\cos(\theta_r - 2\pi/3)$$

$$L_{cD} = L_{Dc} = L_{sD}\cos(\theta_r + 2\pi/3)$$

$$L_{aQ} = L_{Qa} = -L_{sQ}\sin\theta_r$$

$$L_{bQ} = L_{Qb} = -L_{sQ}\sin(\theta_r - 2\pi/3)$$

$$L_{cQ} = L_{Qc} = -L_{sQ}\sin(\theta_r + 2\pi/3)$$

The electromagnetic torque can be expressed as

$$T_e = \left(\frac{P}{2}\right)[i_{abc}]^T \frac{\partial[L_{sr}]}{\partial\theta}[i_{FDQ}] \tag{4.8}$$

where P is the number of poles.

Transformation of state equation to d,q,0 frame

The above resultant differential equations describing the behaviour of electrical machines having time-varying coefficients, which raise difficulties in the simulation. A great simplification can be made by the transformation of stator variables from phases a, b, and c into new variables at a common reference frame. If θ is chosen as the angle difference between the reference frame and the magnetising axis of phase a of the stator winding, then the following transformation matrix can be used:

$$[T_s] = \frac{2}{3}\begin{bmatrix} \cos(\theta) & \cos(\theta-120^\circ) & \cos(\theta-240^\circ) \\ \sin(\theta) & \sin(\theta-120^\circ) & \sin(\theta-240^\circ) \\ 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \tag{4.9}$$

Applying the above transformation to the voltage equation (4.4), yields the following equation on the new reference frame:

$$[T_s]\cdot[V_{abc}]=[T_s]\cdot[R_s][i_{abc}]+[T_s]\cdot\frac{d}{dt}\left([T_s]^{-1}[T_s][\lambda_{abc}]\right)$$

$$[V_{qd0}]=[R_s][i_{qd0}]+[T_s]\cdot\frac{d}{dt}\left([T_s]^{-1}[\lambda_{qd0}]\right)$$

$$[V_{qd0}]=[R_s][i_{qd0}]+[T_s]\cdot\frac{d[T_s]^{-1}}{dt}[\lambda_{qd0}]+\frac{d[\lambda_{qd0}]}{dt}$$

The flux equation can also be transformed:

$$\begin{bmatrix}\lambda_{abc}\\ \lambda_{FDQ}\end{bmatrix}=\begin{bmatrix}T_s^{-1} & 0\\ 0 & U\end{bmatrix}\begin{bmatrix}\lambda_{qd0}\\ \lambda_{FDQ}\end{bmatrix} \tag{4.10}$$

Substituting in (4.6) and (4.7), we have

$$\begin{bmatrix}T_s^{-1} & 0\\ 0 & U\end{bmatrix}\begin{bmatrix}\lambda_{qd0}\\ \lambda_{FDQ}\end{bmatrix}=\begin{bmatrix}L_{ss} & L_{sr}\\ L_{rs} & L_{rr}\end{bmatrix}\begin{bmatrix}T_s^{-1} & 0\\ 0 & U\end{bmatrix}\begin{bmatrix}i_{qd0}\\ i_{FDQ}\end{bmatrix}$$

then

$$\begin{bmatrix}\lambda_{qd0}\\ \lambda_{FDQ}\end{bmatrix}=\begin{bmatrix}T_s & 0\\ 0 & U\end{bmatrix}\begin{bmatrix}L_{ss} & L_{sr}\\ L_{rs} & L_{rr}\end{bmatrix}\begin{bmatrix}T_s^{-1} & 0\\ 0 & U\end{bmatrix}\begin{bmatrix}i_{qd0}\\ i_{FDQ}\end{bmatrix} \tag{4.11}$$

Where

$[V_{qd0}]=[V_q\ V_d\ V_0]^T$, $[i_{qd0}]=[i_q\ i_d\ i_0]^T$ and $[\lambda_{qd0}]=[\lambda_q\ \lambda_d\ \lambda_0]^T$ are qd0 voltages, currents and flux-linkages of stator windings.

Substituting for T_s, T_s^{-1} and the inductance given by the equations above, the equation (4.11) can be simplified. The inductance coefficients become constants and the d axis variables are de-coupled with q axis variables.

The synchronous machine can now be modelled with the following equations:

Voltage equations:

$$\begin{aligned}V_d &= r_s i_d+\frac{d\lambda_d}{dt}-\omega_r\lambda_q\\ V_q &= r_s i_q+\frac{d\lambda_q}{dt}+\omega_r\lambda_d\\ V_F &= r_F i_F+\frac{d\lambda_F}{dt}\\ V_Q &= r_Q i_Q+\frac{d\lambda_Q}{dt}\end{aligned} \tag{4.12}$$

$$V_D = r_D i_D + \frac{d\lambda_D}{dt}$$

and the flux-linkage equations:

$$\lambda_d = L_{ls} i_d + L_{md}(i_d + i_F + i_D)$$

$$\lambda_q = L_{ls} i_q + L_{mq}(i_q + i_Q)$$

$$\lambda_F = L_{lF} i_F + L_{md}(i_d + i_F + i_D) \qquad (4.13)$$

$$\lambda_D = L_{lD} i_D + L_{md}(i_d + i_F + i_D)$$

$$\lambda_Q = L_{lQ} i_Q + L_{mq}(i_q + i_Q)$$

$$L_{md} = \frac{2}{3}(L_{os} + L_{2s})$$

$$L_{mq} = \frac{2}{3}(L_{os} - L_{2s})$$

Electrical torque and motion equations:

$$T_e = \left(\frac{3}{2}\right)\left(\frac{P}{2}\right)(\lambda_d i_q - \lambda_q i_d) \qquad (4.14)$$

$$J\frac{d\omega_r}{dt} = T_e - T_d - \omega_r K_d \qquad (4.15)$$

In the equations, the following notations are made:

λ_d, λ_q	d and q axis stator flux-linkages respectively,
λ_F, λ_D, λ_Q	field, direct, quadrature circuit flux-linkages respectively,
V_d, V_q	d and q axis stator voltages,
V_F, V_D, V_Q	field, direct, quadrature circuit voltages,
L_{ls}	stator leakage inductance,
L_{md}, L_{mq}	d and q axis magnetising inductance respectively,
L_{lF}, L_{lD}, L_{lQ}	field, direct, quadrature circuit leakage inductance respectively,
r_s	stator resistance,
r_F, r_D, r_Q	field, direct, quadrature circuit resistance respectively,
ω_r	rotor angular speed of rotation respectively,
T_e, T_d	electromagnetic and load torque respectively,
K_d	damping constant,
P	number of poles.

The above parameters and equations related to the rotor windings are referred to the stator by the turns ratio transformation.

4.2.2. The permanent magnet generator and its modelling

Features of permanent magnet machines

In conventional synchronous generators, both stator and rotor windings are connected to the external circuits. However, in PM machines, the excitation or field winding is replaced by permanent magnets and, of course, no external source of electrical energy is required. In other respects, a PM machine may be directly compared to conventional synchronous machines, and the armature windings and magnetic circuit in PM machines may be identical to those in conventional machines.

However, there are several major differences between PM machines and conventional synchronous machines:

- *Control.* A major characteristic of conventional synchronous machines is the ability to control various external machine characteristics, such as terminal voltage and/or power factor. This feature is absent in a PM machine, although there have been certain attempts to exercise the same type of control by varying the machine air gap (in axial air gap machines) or by magnetically shunting the field magnets. PM machines have not been considered for central power station alternator applications because of this limitation. However, for relatively small power applications, PM alternators can be operated stand-alone or effectively in parallel with power systems provided properly designed semiconductor switching devices are employed. Such applications include wind power and power generated from waste heat.
- *Cost.* PM machines appear inherently less expensive than wound-field machines due to the simplicity of the permanent magnet versus a field winding excitation. This is not necessarily true, and each case must be evaluated on its own merits and specifications.

- *Volume and Weight.* A PM machine will generally offer volume and weight savings compared to the equivalent wound-field configuration.
- *Flexibility of size and shape.* One of the major merits of PM machines compared to conventional machines is that the PM machine can be constructed in many non-standard sizes and shapes, which often compensates for a cost penalty, especially in automotive and aerospace applications. Machines using PMs with very high residual flux density can be constructed in what is called the "iron-less stator" configuration; that is that the magnetic material in the armature can be eliminated, resulting in a large weight savings. The same type of PMs also permit a machine to have a larger air gap. Both of these potential design features can have beneficial effects on machine performance characteristics, such as cogging and ripple in the air gap flux density, as well as simplifying machine assembly and lowering initial cost.
- *Demagnetisation.* One limitation of a PM machine is that the permanent magnets may be demagnetised by excessive armature reaction, excessive temperatures, or excessive mechanical shock. Considering armature current surges, provisions for frequent magnetisation of alnico magnets is required. However with ceramic and rare-earth magnets, demagnetisation is less of a problem but still a possibility which must be considered. This is probably the principal reason that PM alternators have not been widely adopted in automotive applications. Recently, many innovative configurations have been developed which divert the demagnetising armature reaction from the path of the PMs during current surges.

PM synchronous machines generally have the same operating and performance characteristics as synchronous machines: operation at synchronous speed, a single-phase or poly-phase source of alternating current flowing in the armature windings, a power limit above which operation at synchronous speed is unstable, and damper (cage) windings for starting and stability purpose. A PM machine can have a configuration almost identical to that of the conventional synchronous machine with

the absence of brushes, slip rings and a field winding. The inability to control fluxing is responsible for the one major difference between a PM machine and conventional synchronous machine, the lack of power factor or reactive power control. Consequently terminal voltage regulation represents a challenge.

The absence of brushes, slip rings and field winding is a principal merit of the PM synchronous machine. However, it is the low cost, simplicity of rotor construction, and reduced manufacturing assembly procedures and costs of a PM machine that makes it very competitive in the market. On the basis of power output per unit weight (and, in general, per unit volume), the PM synchronous machine is superior to all other brush-less synchronous machines.

Mathematical equations

The modelling equations for a permanent magnet generator can be easily obtained by simplify the synchronous machine equations. In general, many types of permanent magnet machines exist which can be represented by using different constraints on Park's equations. The most general type is obtained by assuming that permanent magnet excitation is simply equivalent to a constant field current i_{Fc} in Park's equations. Then defining the constant flux-linkage

$$\Lambda_F = L_{md} i_{Fc} \tag{4.16}$$

Park's equations for a permanent magnet machine become

$$\begin{aligned} V_d &= r_s i_d + \frac{d\lambda_d}{dt} - \omega_r \lambda_q \\ V_q &= r_s i_q + \frac{d\lambda_q}{dt} + \omega_r \lambda_d \\ V_Q &= r_Q i_Q + \frac{d\lambda_Q}{dt} \\ V_D &= r_D i_D + \frac{d\lambda_D}{dt} \end{aligned} \tag{4.17}$$

and the flux-linkage equations:

$$\begin{aligned}\lambda_d &= L_{ls} i_d + L_{md}(i_d + i_D) + \Lambda_F \\ \lambda_q &= L_{ls} i_q + L_{mq}(i_q + i_Q) \\ \lambda_D &= L_{lD} i_D + L_{md}(i_d + i_D) + \Lambda_F \\ \lambda_Q &= L_{lQ} i_Q + L_{mq}(i_q + i_Q)\end{aligned} \tag{4.18}$$

Electrical torque and motion equations are the same as that of the synchronous machine.

4.3. Generator simulation

Simulation models for synchronous machines have been developed in Matlab/Simulink. The simulation assumes a linear magnetic circuit with no saturation effects in the stator and rotor core. This assumption can be made because of the large air gap usually found in permanent magnet synchronous machines.

The block diagrams of a field excited machine are shown in Figures 4.3, 4.4, 4.5, 4.6 and 4.7. Figure 4.3 is the entire model of the field excited machine. Figures 4.4 and 4.5 give the d axis and q axis models respectively. Figure 4.6 illustrates the torque model while Figure 4.7 shows the motion model.

Figure 4.8 shows the speed curve during acceleration of a field excited synchronous machine. The parameters of this machine are:

$$r_s = 0.1\Omega \qquad r_{fd} = 0.016\Omega \qquad r_{kd} = 0.17\Omega \qquad r_{kq} = 0.17\Omega$$

$$L_{md} = 4.1mH \qquad L_{mq} = 2.0mH \qquad L_{ds} = 4.89mH$$

$$L_{qs} = 2.79mH \qquad L_{fq} = 4.48mH \qquad L_{kd} = 4.39mH$$

$$L_{kq} = 2.91mH \qquad J = 0.2Nms^2$$

Figure 4.9 and 4.10 show the block diagrams illustrating the models for the permanent machine. Due to the similarity between the field excited and permanent magnet machine, the models of q axis, torque, and motion for permanent magnet machine are the same as those models of the field excited machine. A constant current is used to

simulate the field established by the permanent magnet in Figure 4.10. The parameters used for the permanent magnet machine are as follows:

$$r_s = 0.32\Omega \qquad r_{kd} = 0.99\Omega \qquad r_{kq} = 2.0\Omega$$

$$L_{md} = 23mH \qquad L_{mq} = 50mH \qquad L_{ds} = 26.2mH$$

$$L_{qs} = 53.2mH \qquad L_{kq} = 56.4mH \qquad L_{kd} = 29.4mH$$

$$J = 0.01Nms^2$$

The machine speed acceleration curve for the field excited machine is shown in Figure 4.8 where a constant driving torque of 20 Nm is applied. While the acceleration curve of permanent magnet machine is given in Figure 4.11, the driving torque to the PM machine is 18 Nm. In both cases, machines are loaded by pure resistive load (R_d=R_q= 100 Ω). The PM machine shows a faster acceleration due to the smaller machine inertia.

The other important aspect of a stand-alone generator system is the power electronic system. The next chapter deals with the basics of power electronics and concludes with simulation models for the dc-dc converter and relevant control systems.

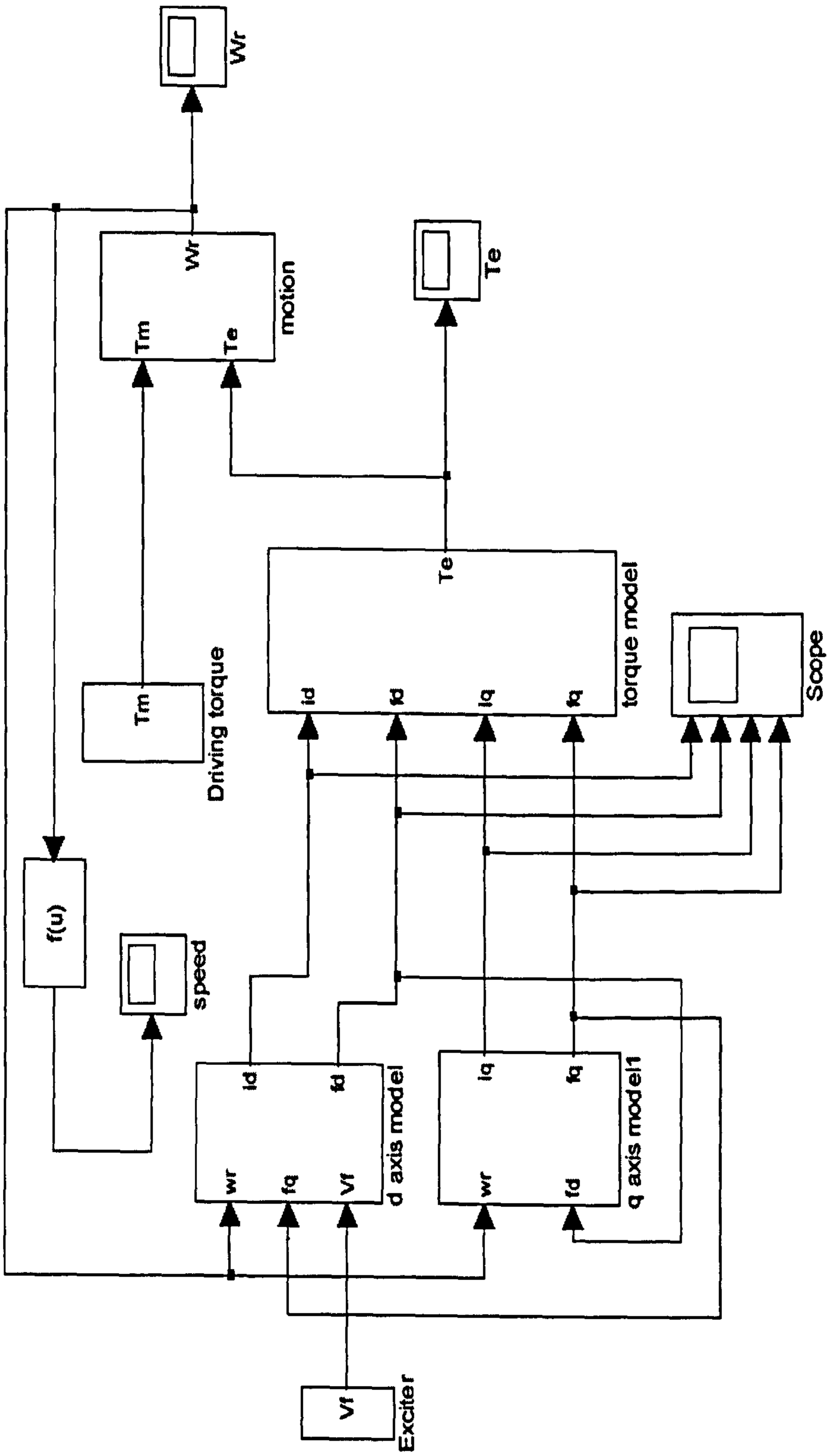


Figure 4. 3. Synchronous machine simulation model

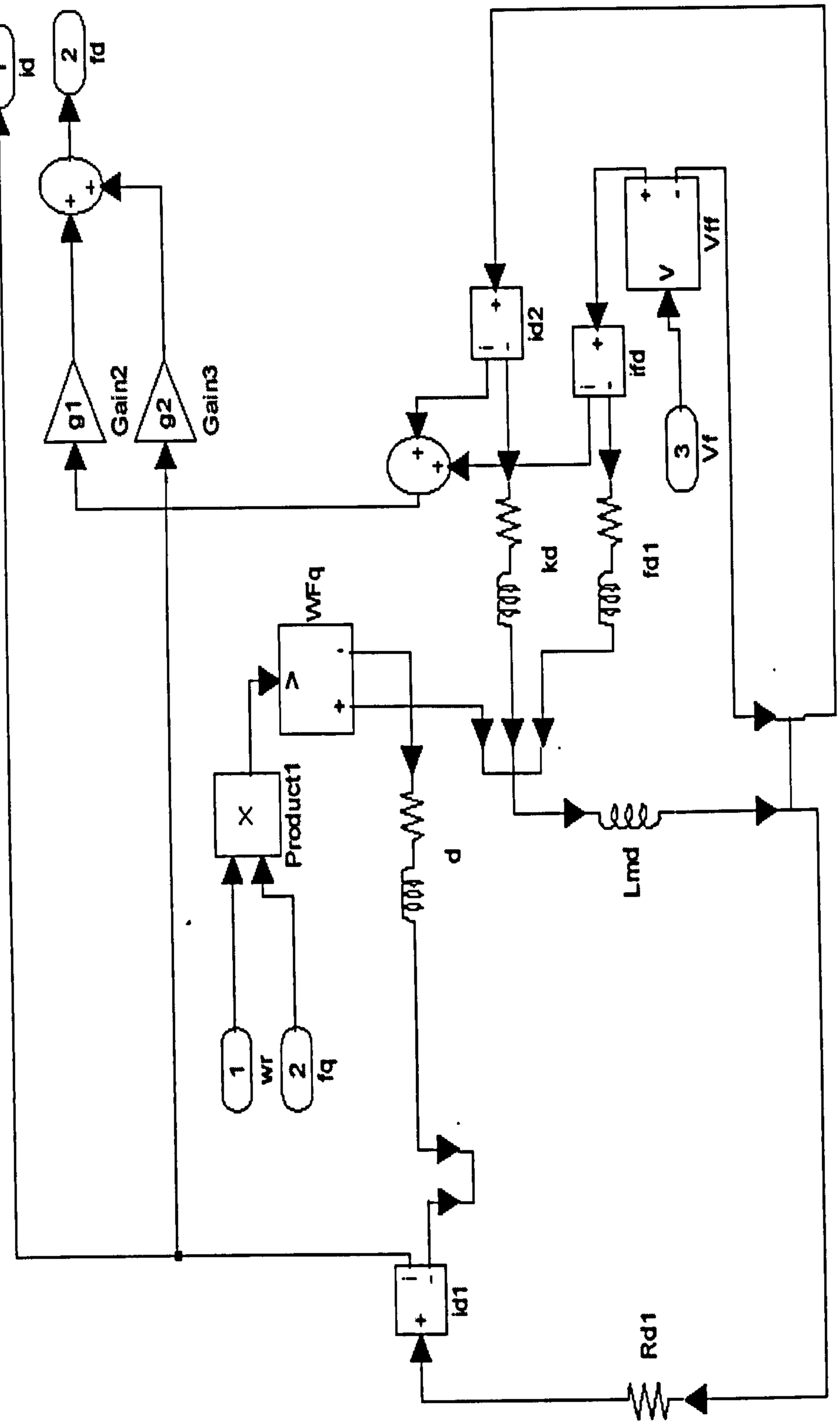


Figure 4. 4. Synchronous machine d axis simulation model

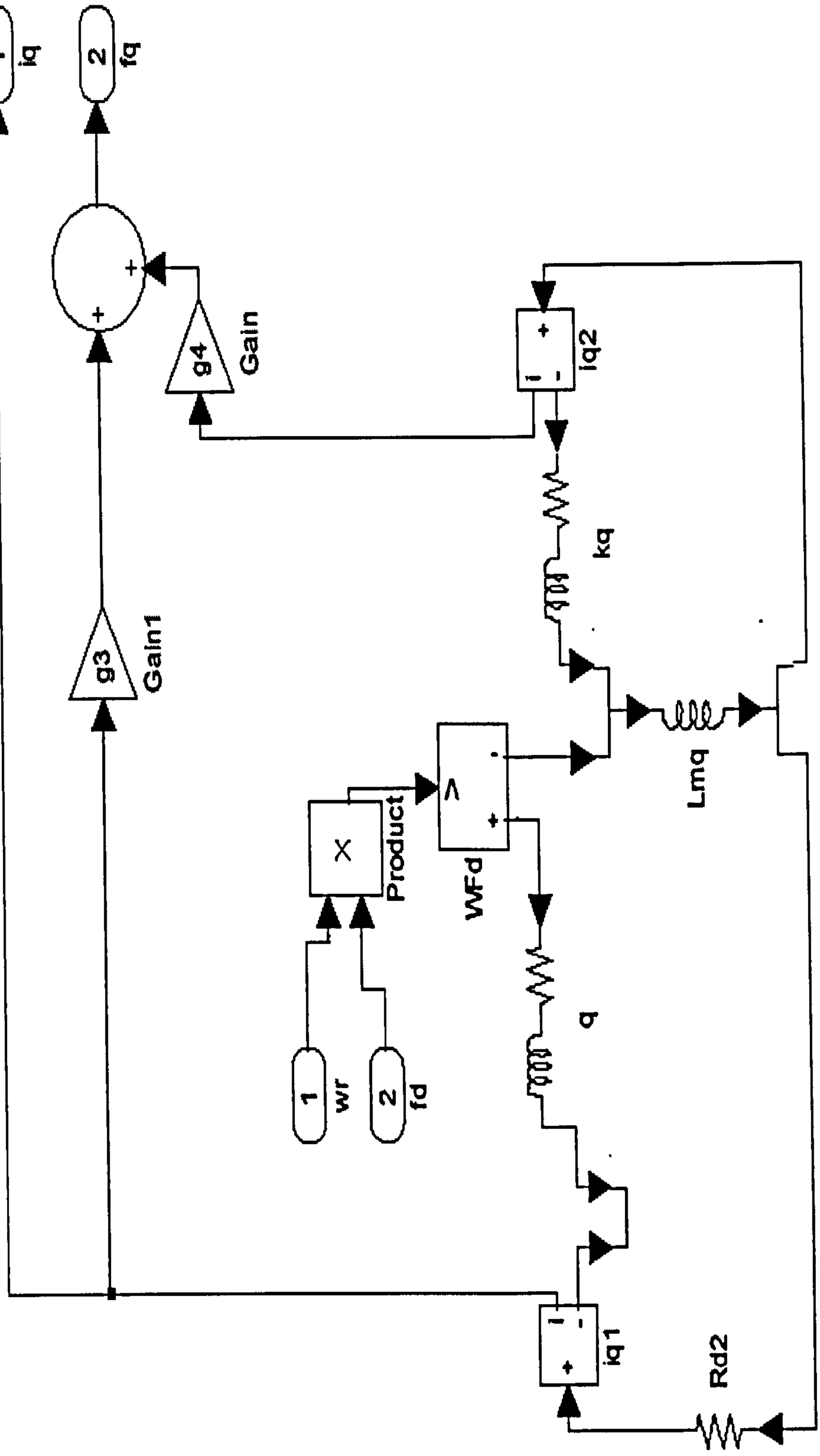


Figure 4. 5. Synchronous machine q axis simulation model

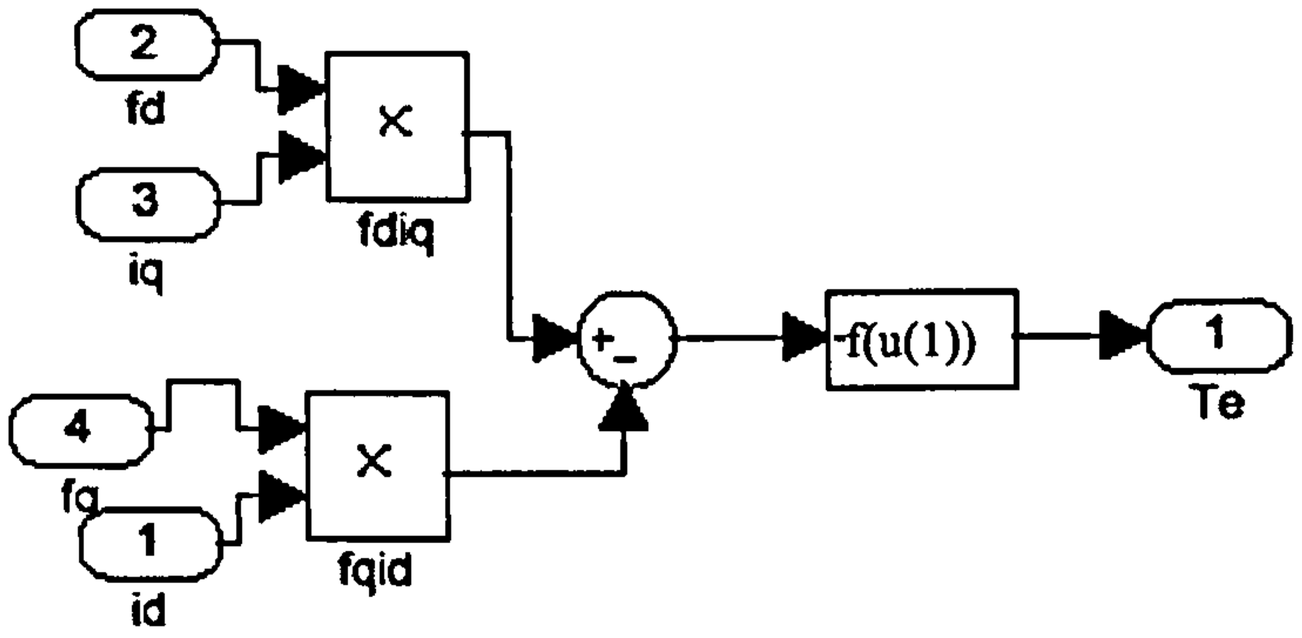


Figure 4.6. Synchronous machine torque model

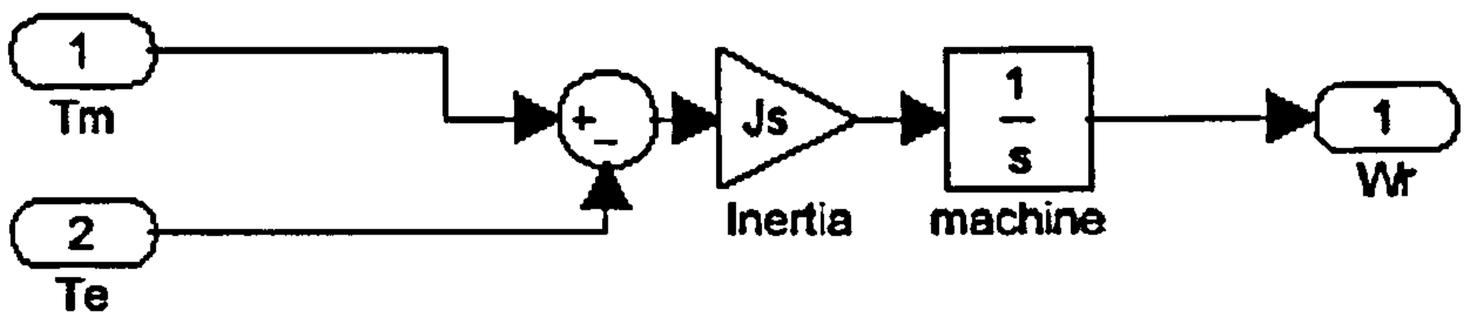


Figure 4.7. Synchronous machine motion model

Figure 4.8. Speed curve during acceleration (SM)

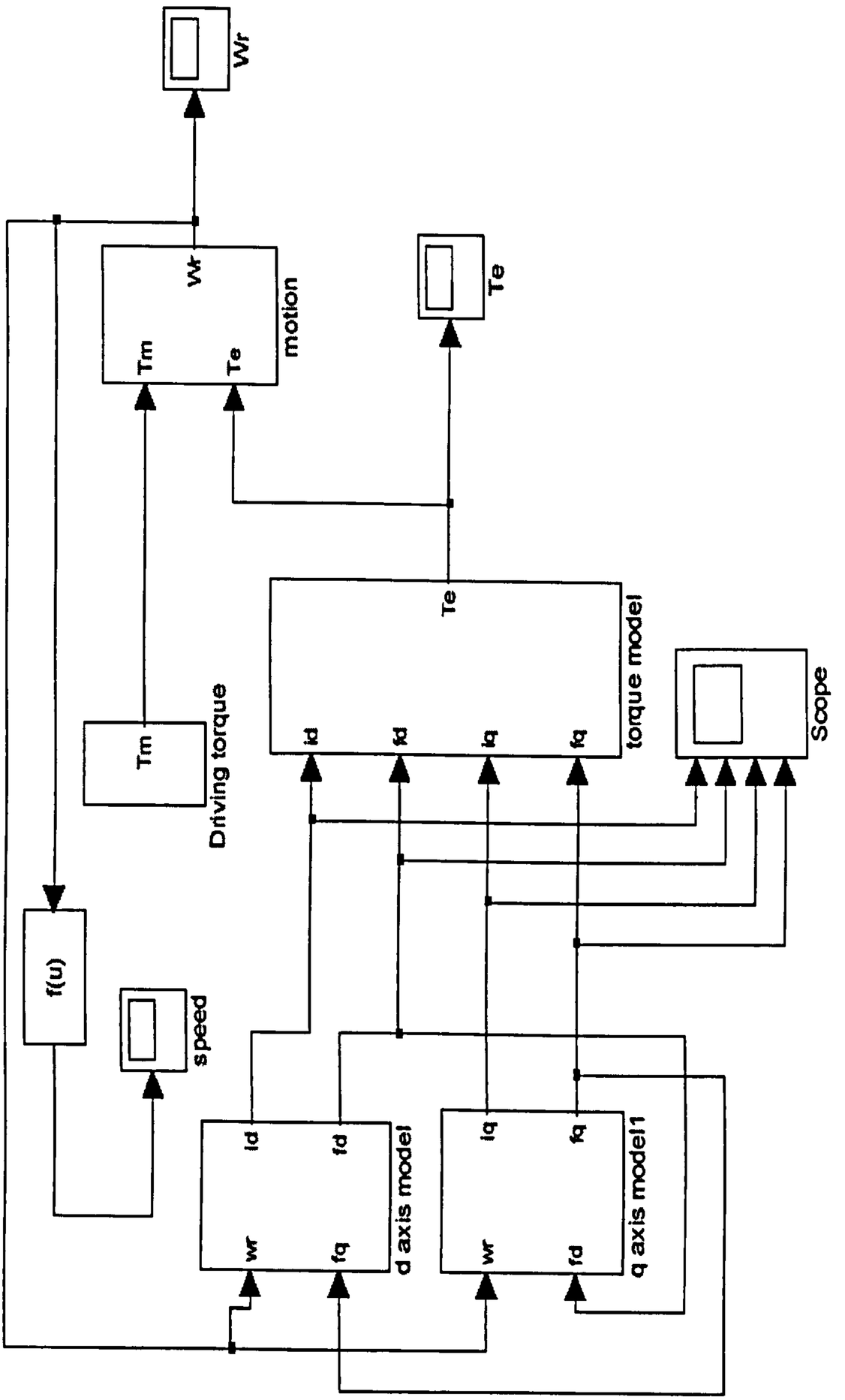


Figure 4.9. Permanent machine simulation model

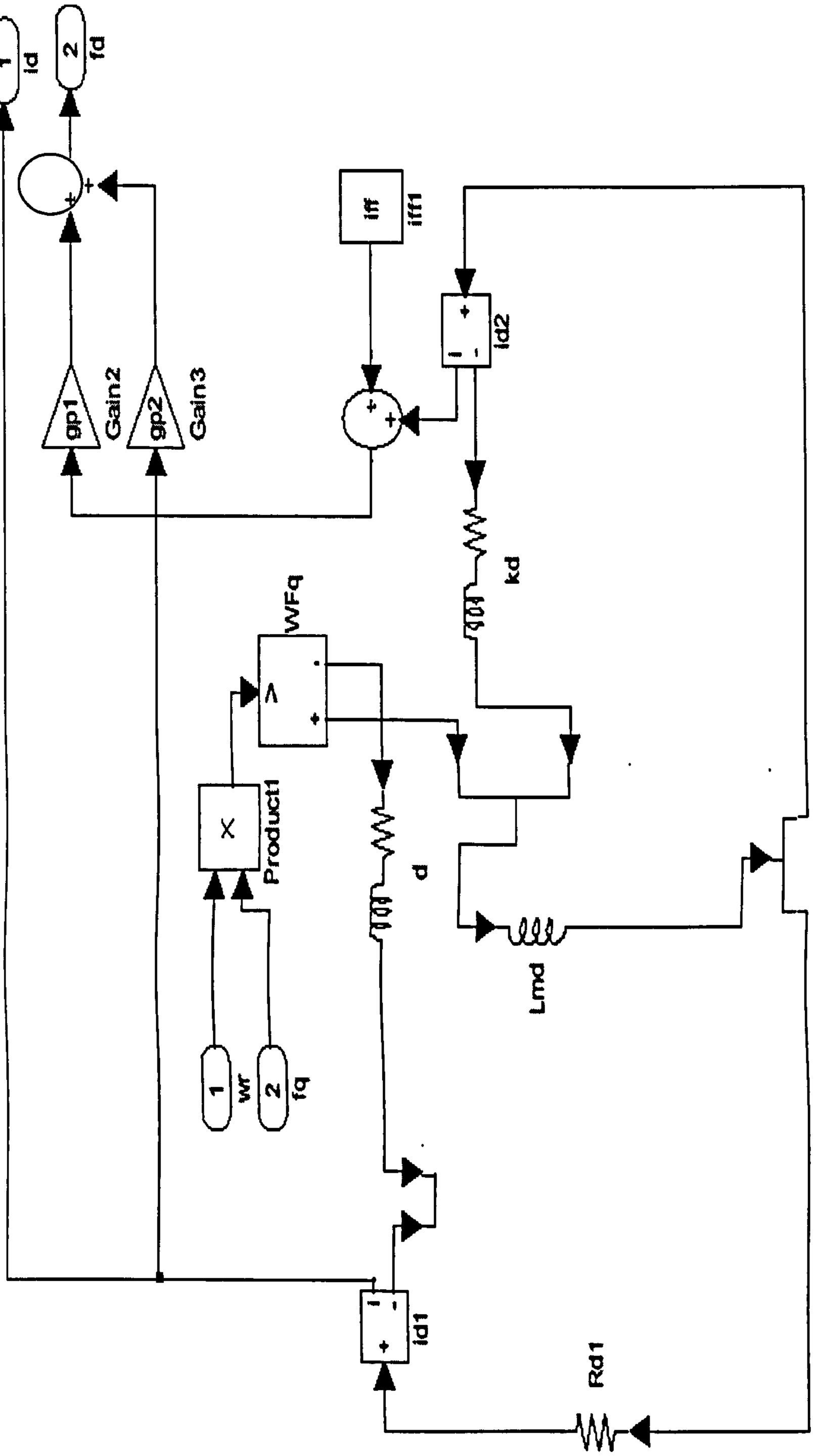


Figure 4.10. Permanent magnet machine d axis simulation model

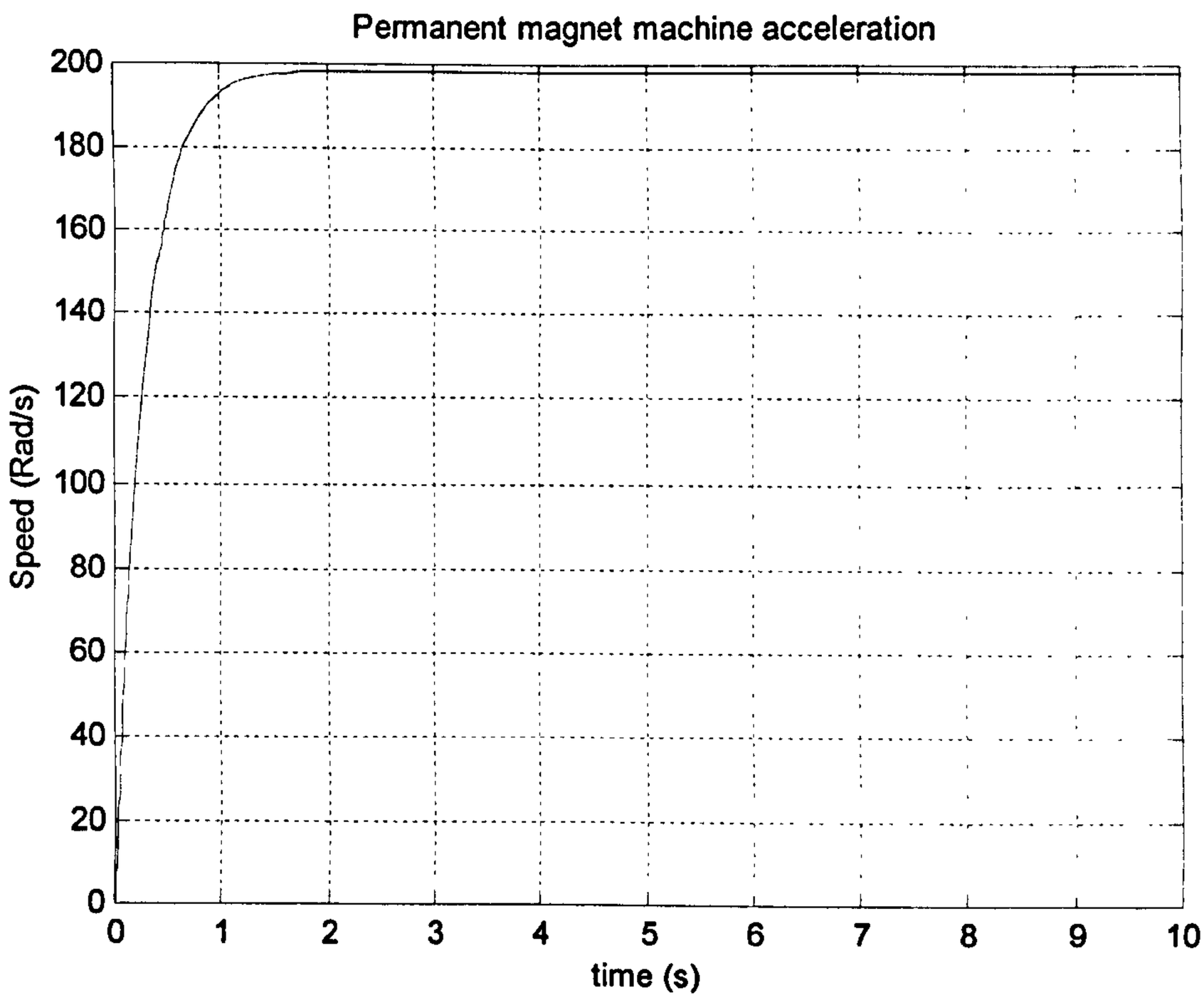


Figure 4.11. Speed curve during acceleration (PM)

Chapter 5

The Power Electronic System

This chapter discusses the power electronic systems used in this project, namely inverters, dc-dc converters and their associated switching strategies. Inverters are discussed, the suitable PWM switching methods for inverters, Space Vector Modulation (SVM) methods, are addressed in detail. Then dc-dc converters are presented together with their switching method. The simulation models for the chosen power electronic converters are developed. Simulation investigations are carried out to verify the developed models and switching strategies and to study the performance of the converters.

5.1. Power electronic inverters

5.1.1. Basic inverter configurations

There are basically two types of power electronic converters, voltage source and current source types. Voltage Source Inverter (VSI) and Current Source Inverter (CSI) are the corresponding dc-ac converter circuits widely used in various power electronic applications [46][49][52][29][18][47]. For a voltage source inverter, the combination of a rectifier and a dc link capacitor behaves like a voltage source to the inverter so that a three-phase VSI will produce a three-phase ac PWM voltage waveform. For a current source inverter, the combination of a rectifier and a dc link inductance appears as a current source to the inverter, therefore a CSI produces a switched current waveform at its output. The basic three phase CSI and VSI configurations are shown in Figure 5.1 and Figure 5.2 respectively.

Duality exists between VSI and CSI systems [34]. The main characteristics of the two types of inverter systems are:

VSI System

- Output is constrained voltage
- DC bus is supported by shunt capacitor
- Produces voltage harmonics
- Prefers load with large reactance
- DC bus current reverses in regeneration
- Immune to open circuits

CSI System

- Output is constrained current
- DC bus is dominated by series inductor
- Generates current harmonics
- Prefers motors with lower reactance
- DC bus voltage reverses in regeneration
- Immune to short circuits

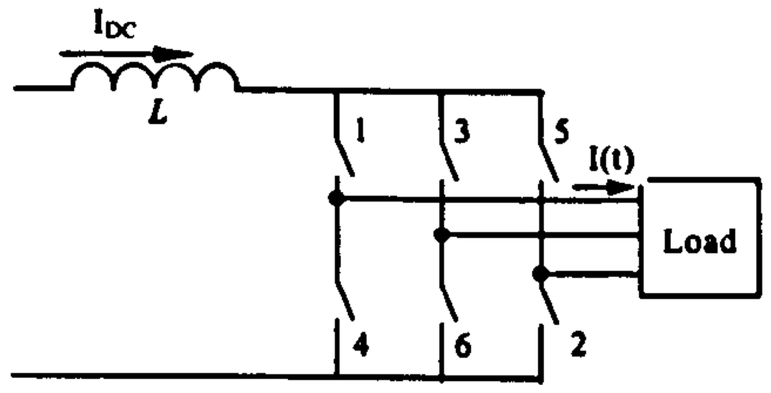


Figure 5.1. A basic CSI configuration

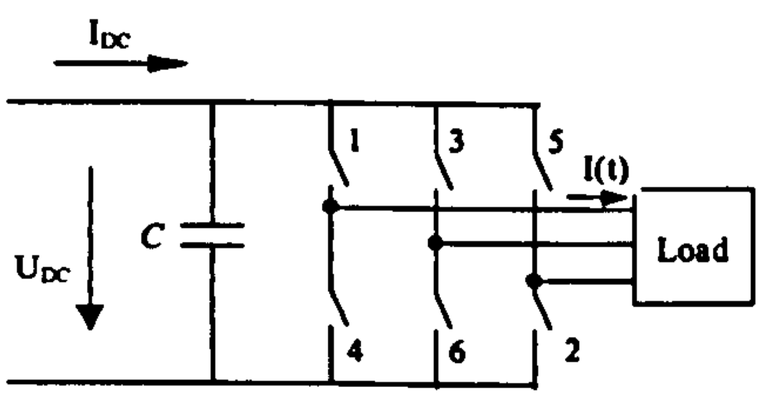


Figure 5.2. A basic VSI configuration

Various PWM schemes are used for power electronic converter applications. For example, conventional sinusoidal PWM employs different sampling methods according to a carrier signal, such as natural sampling or regular sampling (symmetric or asymmetric), modified sine PWM, harmonic injection PWM, programmed harmonic elimination, etc. [6][25][19][28]. However, the space vector PWM approach has the advantage of higher efficiency over other methods. Therefore, the space vector

PWM methods for voltage source and current source inverters will be discussed in this chapter to establish a suitable inverter control strategy for the studied stand alone diesel engine generator system.

5.1.2. The Space Vector PWM technique for CSI

The phasor diagram used to describe the space vector modulation (SVM) technique in a CSI configuration is illustrated in Figure 5.3 where I_{ref} is the ideal current needed to be generated by the inverter [52].

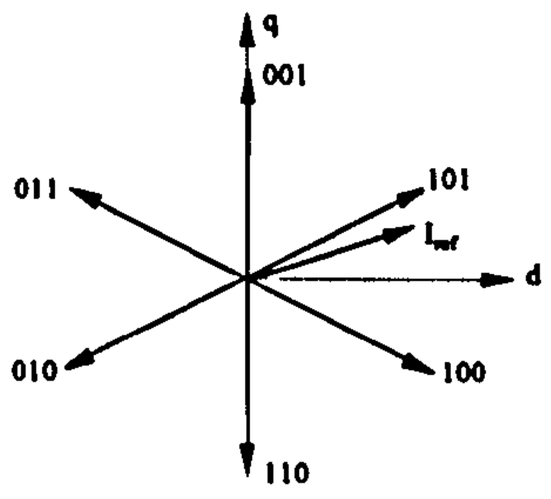


Figure 5.3. CSI output phasor diagram

Considering the operation of a CSI, there are nine possible switching positions as shown in Figure 5.4.

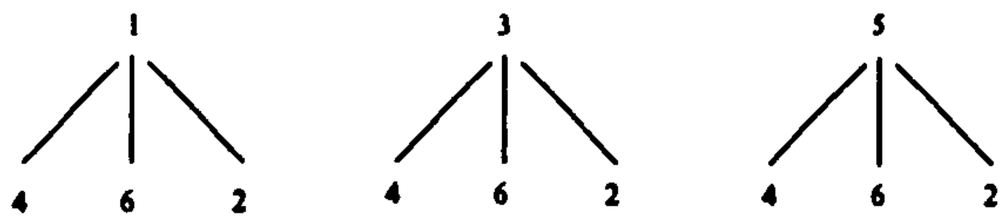


Figure 5.4. Switching position

Referring to Figure 5.1, the combinations of switches 1-4, 6-3, 5-2 (the two devices on the same inverter leg are fired at the same time) produce zero output current.

The reference current in Figure 5.3 can be approximated by combining certain switch states shown in Figure 5.5.

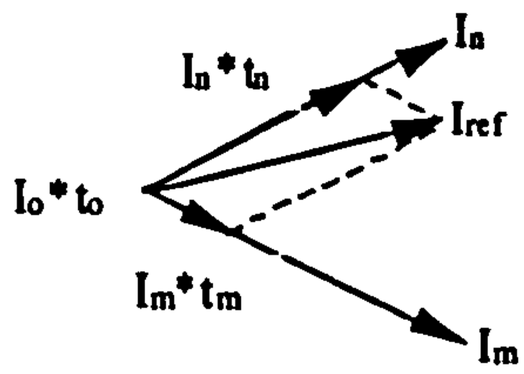


Figure 5.5. Explanation of parameters

To produce the required current space vector, the PWM strategy can be described as follows

$$I_{ref} \times T_s = I_m \times t_m + I_n \times t_n + I_0 \times t_0 \qquad (5.1)$$

All parameters in equation (5.1) are illustrated in Figure 5.5 where t_m, t_n and t_o are the respective time lengths corresponding to vectors I_m, I_n and I_o.

In the CSI output phasor diagram, assuming a non-power invariant transformation, any current vector I_p can be represented by

$$I_p = \sqrt{\frac{2}{3}} I_{dc} * F_s \qquad (5.2)$$

$$F_s = \sqrt{3} e^{j\frac{\pi}{3}(m-\frac{1}{2})} \qquad m=1,\ldots,6. \qquad (5.3)$$

$$F_s = 0 \qquad m=0$$

To calculate the time intervals, t_m, t_n, and t_o, it is convenient to represent the reference phasor I_{ref} in an arbitrary sextant k shown in Figure 5.6.

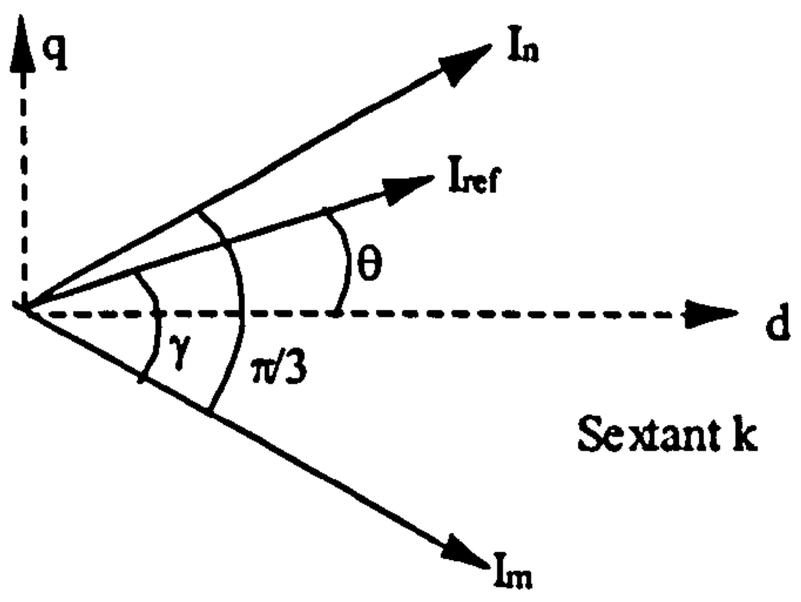


Figure 5.6. Current I_{ref} in sextant k

The current in Figure 5.6 can be expressed as

$$I_m = \sqrt{3}|I_m| e^{j((k-1)\frac{\pi}{3}-\frac{\pi}{6})} \qquad (5.4)$$

$$I_n = \sqrt{3}|I_n| e^{j(k\frac{\pi}{3}-\frac{\pi}{6})} \qquad (5.5)$$

where

$$|I_m| = |I_n| = \sqrt{\frac{2}{3}} I_{DC}$$

$$I_{ref} = |I_{ref}| e^{j\theta} \tag{5.6}$$

θ is measured from the stationary axis d as

$$\theta = \gamma - \frac{\pi}{6} + (k-1)\frac{\pi}{3}$$

Equation (5.7) can be derived by substituting (5.4), (5.5), (5.6) into (5.1) and considering that I_0 is equal to zero:

$$|I_{ref}| e^{j\theta} T_s = \sqrt{\frac{2}{3}} I_{DC} (\sqrt{3} e^{j((k-1)\frac{\pi}{3} - \frac{\pi}{6})} t_m + \sqrt{3} e^{j(k\frac{\pi}{3} - \frac{\pi}{6})} t_n \tag{5.7}$$

The equation above can be separated into its real and imaginary parts:

$$\frac{T_s M_I}{\sqrt{3}} = t_m \cos\gamma + t_n \cos(\frac{\pi}{3} - \gamma) \tag{5.8}$$

$$0 = -t_m \sin\gamma + t_n \sin(\frac{\pi}{3} - \gamma) \tag{5.9}$$

where

$M_I = \dfrac{I_{ref}}{\sqrt{\frac{2}{3}} I_{DC}}$ is the modulation ratio.

Solving equations (5.8) and (5.9), t_m and t_n can be obtained as:

$$t_m = \frac{2}{3} T_s M_I \sin(\frac{\pi}{3} - \gamma) \tag{5.10}$$

$$t_n = \frac{2}{3} T_s M_I \sin(\gamma) \tag{5.11}$$

The time t_0 can then be calculated by

$$\frac{t_0}{2} = \frac{T_s}{2} - \frac{t_m + t_n}{2} \tag{5.12}$$

According to the equations of (5.10), (5.11) and (5.12), the reference current I_{ref} can be approximated by keeping the converter at different current vectors, I_m, I_n and I_0 for time lengths of t_m, t_n and t_0 respectively.

5.1.3. The Space Vector PWM technique for VSI

A space vector PWM for VSI is based on the space vector representation of the voltages in the α, β frame. The α, β components can be found by Park transform of three phase voltages, where the total power remains unchanged. This concept is discussed in references [28][49][46].

The operational principle of the space vector PWM VSI can be clearly explained by using a space vector. The desired voltage vector, V_{des}, can be expressed as a combination of the inverter three phase output voltage v_a, v_b and v_c, which can be expressed in vector form as:

$$\overline{V}_{des} = \sqrt{\frac{2}{3}}(v_a + av_b + a^2 v_c)$$

where

$$a = e^{j120^0}$$

$$v_a = V_m \sin \omega t$$

$$v_b = V_m \sin(\omega t - 120^0)$$

$$v_c = V_m \sin(\omega t + 120^0)$$

and V_m is the amplitude of the fundamental component.

There are eight possible switching positions, as shown in Figure 5.7, in a three-phase PWM voltage source inverter. The corresponding voltage vectors are depicted in Figure 5.8.

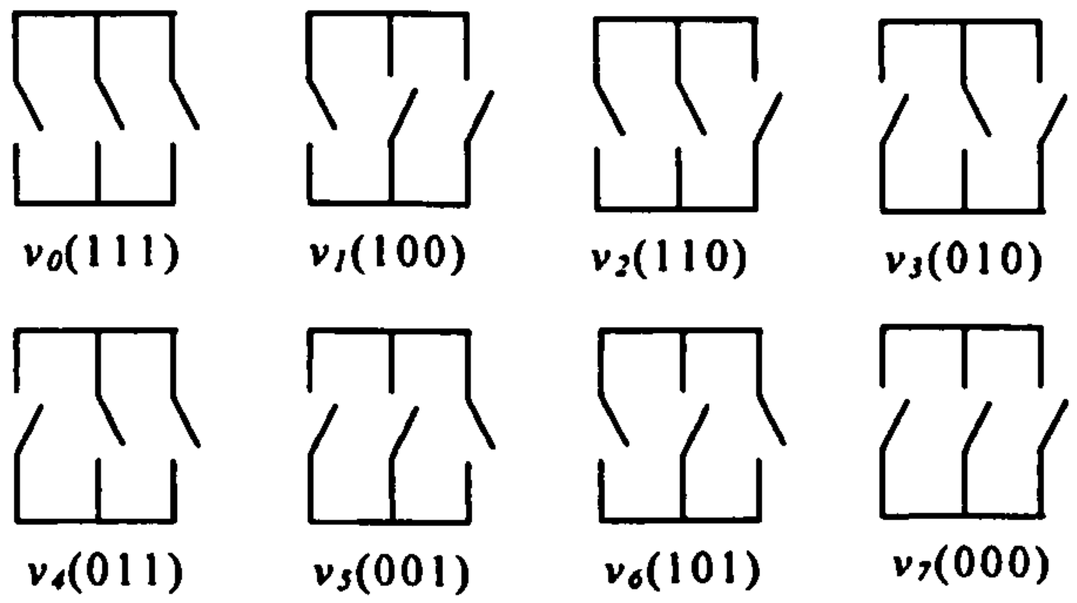
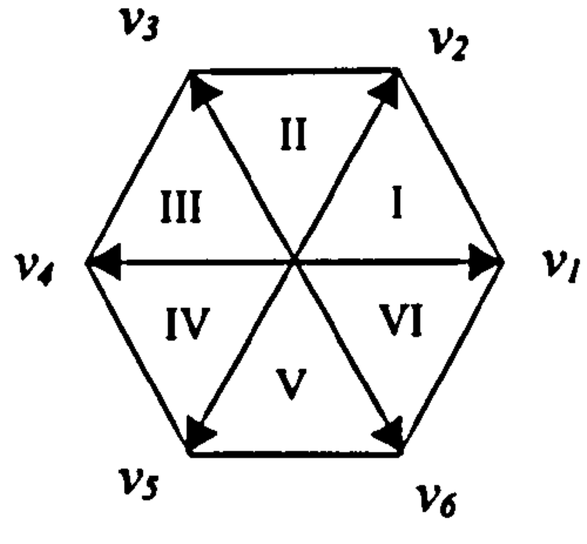


Figure 5.7. Eight positions of a three-phase PWM voltage source inverter



v_0, v_7: zero voltage vector

Figure 5.8. Basic voltage vectors

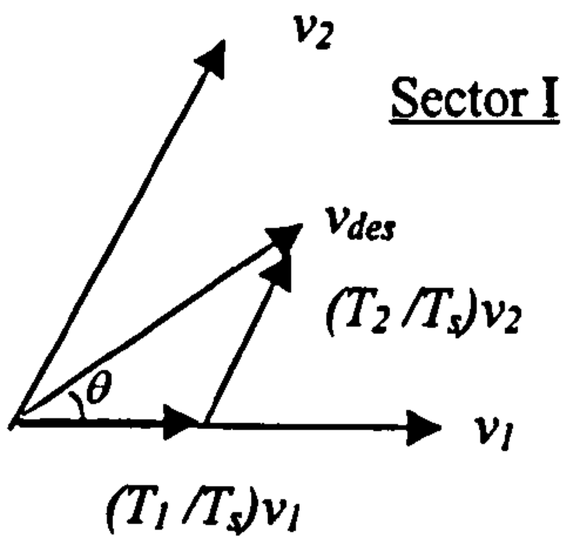


Figure 5.9. Voltage vector decomposition

The desired voltage vector can be decomposed into two components in a two-axis co-ordinate or as a combination of two basic vectors. An example of the voltage vector decomposition is given in Figure 5.9. The space vector PWM strategy for VSIs can aim to minimise harmonic distortion in the current by selecting the appropriate switching vectors and determining their corresponding dwelling periods. If the reference vector is located in sector I as shown in Figure 5.9, then it is composed of voltage vectors of v_1 and v_2, and zero voltage vectors v_0 and v_7. The relationships between the desired voltage vector and its two corresponding basic voltage vectors can be expressed as

$$\overline{V}_{des} = v_1 \frac{T_1}{T_s} + v_2 \frac{T_2}{T_s}$$

where T_s is the period for $\overline{V}_{des}$ and T_1 and T_2 are the dwelling time for v_1 and v_2 respectively. This voltage space vector can be described in rectangular co-ordinates as:

$$T_1 \cdot \sqrt{\frac{2}{3}} V_d \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + T_2 \cdot \sqrt{\frac{2}{3}} V_d \cdot \begin{bmatrix} \cos 60^0 \\ \sin 60^0 \end{bmatrix} = T_s \cdot \sqrt{\frac{2}{3}} V_d \cdot k \cdot \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \qquad (5.13)$$

where $k = \left|V_{des}\right| \Big/ \sqrt{2/3}V_d$, $0 \le \theta \le 60^0$, and V_d is the dc-link voltage. T_1, T_2 and T_0 can be solved from (5.13) [45].

$$T_1 = T_s \cdot k \cdot \frac{\sin(60-\theta)}{\sin 60^0}$$

$$T_2 = T_s \cdot k \cdot \frac{\sin\theta}{\sin 60^0}$$

$$T_7 = T_0 = \frac{T_s}{2} - T_2 - T_1.$$

It is preferred that the cycle time is as short as possible so that the maximum deviation between the reference voltage vector and average voltage formed by switching states minimised. However a higher frequency results in high switching power loss. For a certain sampling frequency, the high performance of PWM can be obtained if

- The cycle normally consists of only three successive switching states representing two switching states adjacent to the reference vector and a zero voltage vector.
- The switching sequence of each inverter leg is such arranged that the transition from one state to another is performed by switching only one inverter switch, in order to minimise switching frequency and power loss.
- The switching of the inverter starts from one zero state and ends at the other zero state.

5.1.4. PWM converter simulation

The VSI strategy is adopted for the system under investigation due to the system and electric equipment requirements. For a stand alone system, when the load is changing, all the parameters related to the load could change, for example, the current flowing into the load side from the inverter will change, and the voltage will also be affected. However, electrical equipment is normally designed to operate at a rated voltage, that means the inverter output voltage, which is applied to the load, should be kept constant.

The power electronic system should be able to condition the output power of the system so as to keep a constant amplitude of the voltage at a fixed frequency while the load varies. The frequency of the ac voltage can then be determined by the load side inverter and the amplitude of the ac voltage can be either controlled by controlling the inverter or a dc-dc converter [10][11][12]. By means of frequency de-coupling via a dc link, the stand-alone system can be efficiently controlled to meet the varying load demand.

In this project, the high efficient space vector PWM method for VSI control is used to provide output voltage with constant frequency and amplitude.

Based on the mathematical analysis, the space vector PWM scheme has been simulated in Matlab. The block diagram of the space vector PWM scheme is shown in Figure 5.10. Figure 5.11 illustrates the ideal line-line voltage waveform of the space vector PWM VSI scheme.

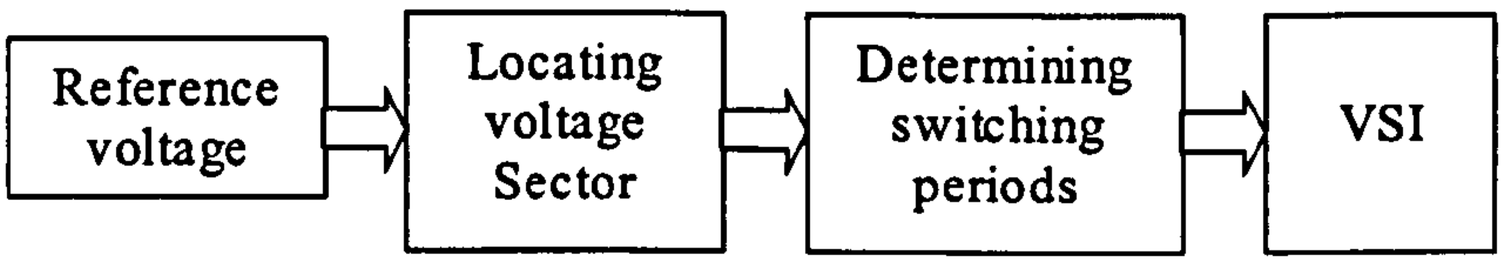


Figure 5.10. Block diagram of space vector PWM scheme

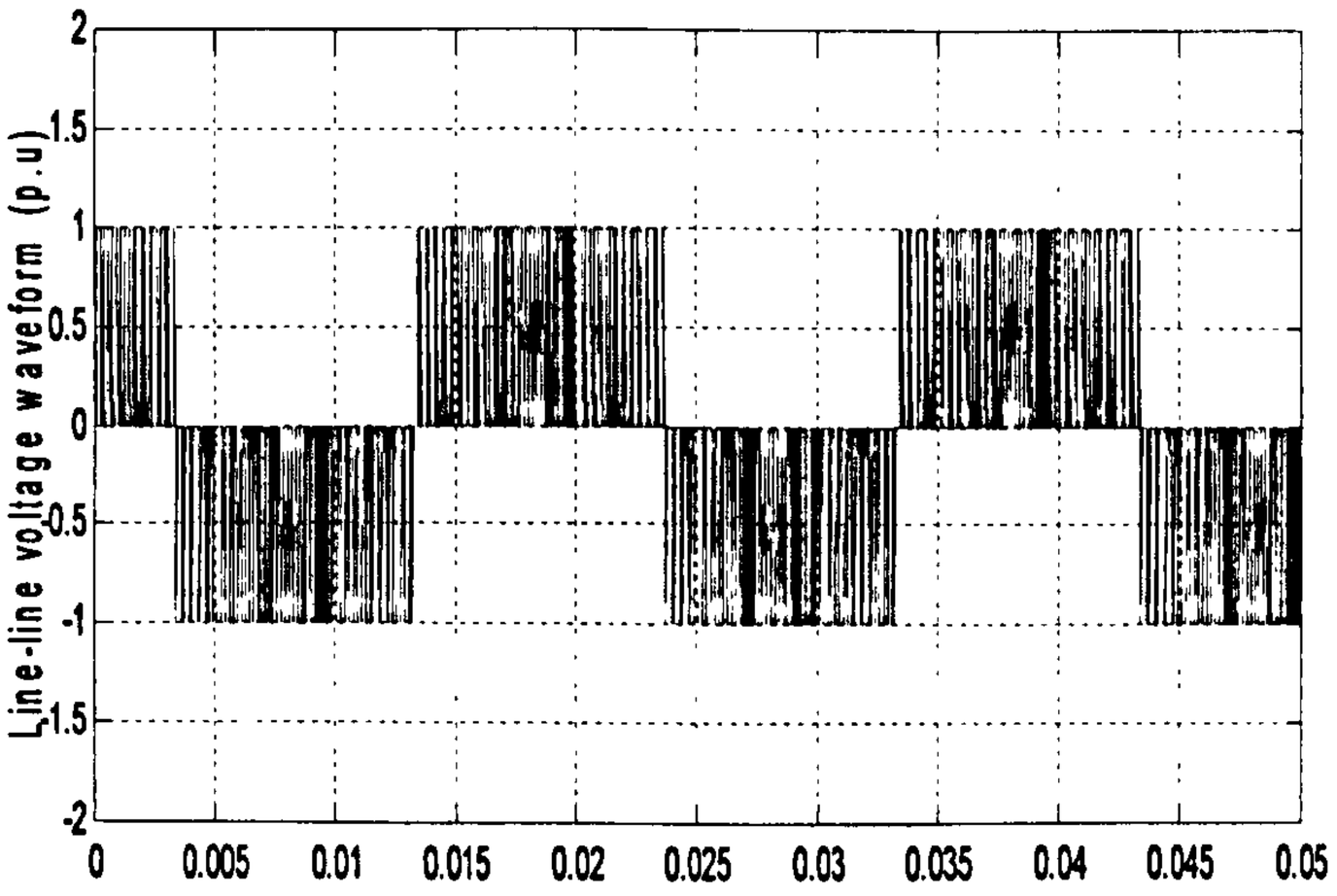


Figure 5.11. Line-line voltage waveform of a space vector PWM scheme

5.2. DC-DC power conversion

DC-DC conversion is an effective way to provide a controllable dc voltage for power electronic systems. In the system discussed, the amplitude of the ac voltage at the inverter output terminal is controlled through a dc-dc converter instead of PWM switching of the inverter. The following sections will discuss dc-dc converters and the associated control method used in the stand alone generation system for maintaining the system voltage level.

5.2.1. DC-DC converter introduction

In ac systems, transformers are used to convert electric power efficiently from one voltage level to another. Static dc-dc converters play a similar role in dc applications. The operation of ac transformers is based on an alternating magnetic field. In dc-dc converters, the voltage conversion is achieved by switching power semiconductors at a high repetitive frequency. Static dc-dc converters are also known as choppers sometimes.

DC-DC converters can be found in many applications. For example, a dc-dc converter can be used on its own for stabilising or controlling a dc voltage from a battery, or it may be an integral part of a more complex converter, controlling an intermediate or an output voltage. DC-DC converters are employed for feeding traction equipment (trucks, electric vehicles, underground locomotives), telephone equipment and inverters for uninterruptable power supply systems or for frequency controlled induction motors. With the increasing developments of renewable energy, such as solar and wind power, dc-dc converters are also found applications in the renewable power generation systems [31][23][12].

In a chopper converter, the voltage conversion ratio is determined by the ratio of ON-state to OFF-state of the static switches. It is therefore easy to vary the voltage conversion ratio smoothly and continuously by suitably modifying the timing of the switching control pulses of the power switching elements. AC power transformers do not normally have such a facility of static control of voltage conversion ratio.

Therefore, the controllability of the voltage conversion ratio statically (without moving contacts) gives the chopper a great usefulness as a power controller.

In this programme, the dc-dc conversion stage is introduced into a stand-alone diesel engine generator system for voltage control. The configuration is shown in Figure 5.12, where the chopper operates at a variable conversion ratio and an almost constant voltage output at the inverter input terminal. In this way the inverter can be operated at the fixed voltage modulation ratio and fixed frequency.

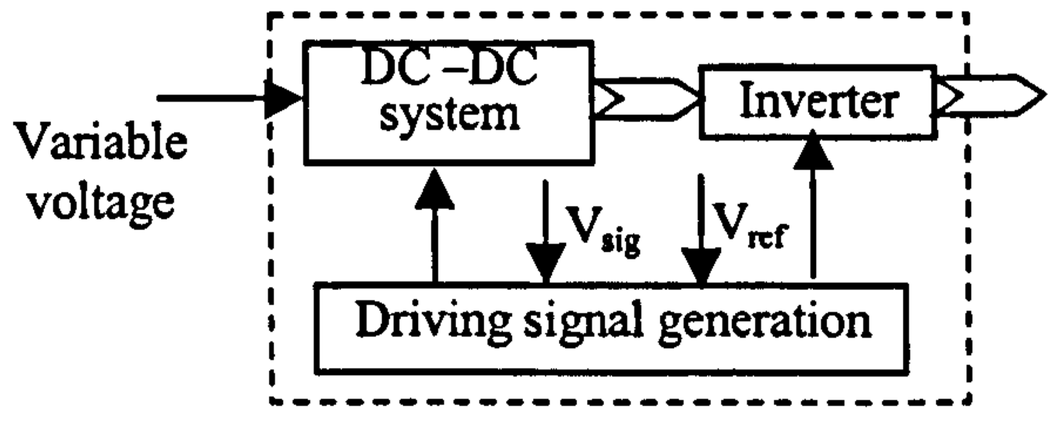


Figure 5.12. Inverter output voltage controlled through a dc-dc conversion stage

5.2.2. Principles of DC-DC converter

Figure 5.13 shows the circuit configuration of a voltage step down chopper. The chopper delivers an adjustable dc power into a resistive load from a dc voltage source. The main task of the dc-dc converter is to convert the unregulated dc input into a controlled dc at a desired voltage level. As the name implies, a step-down converter produces an average output voltage not higher than the input voltage. The basic chopper power circuit is shown boxed with dashed lines in Figure 5.13. It consists of two power semiconductor devices, which function as static switches. The switch labelled S_1 is a controlled switching device, it could be any one of the several types of power semiconductor switching devices. The device should be able to be turned on or turned off by an appropriate control signal on its control terminal. For example, if the device selected for S_1 is a gate turn off thyristor (GTO), it is turned on by a short positive current pulse on its gate terminal and turned off by a short negative current pulse on the same terminal. Alternatively if S_1 is an n-channel power MOSFET, turn on switching is achieved by applying on its gate terminal a positive voltage pulse lasting for the entire duration of its on time. The off-state of the switch is obtained by making the gate voltage zero. The second switch shown in the figure 5.13, labelled S_2,

is actually a power diode. The power diode is a static switch without a control terminal. It automatically turns on whenever the "forward" voltage is available and turns off when the carried current decays to zero. Both static switches are turned on and turned off at a high frequency. To simplify the description of the principle, both S_1 and S_2 will be considered as ideal switches. This implies that (1) their transitions between the on and off states are instantaneous without time delays and (2) the on state voltage drop across each switch is zero.

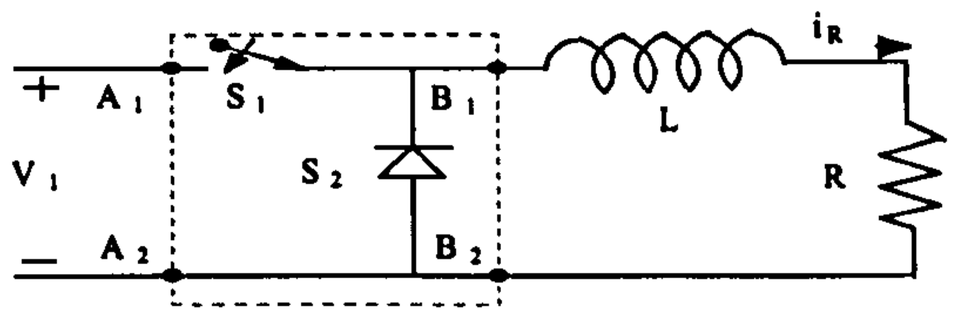


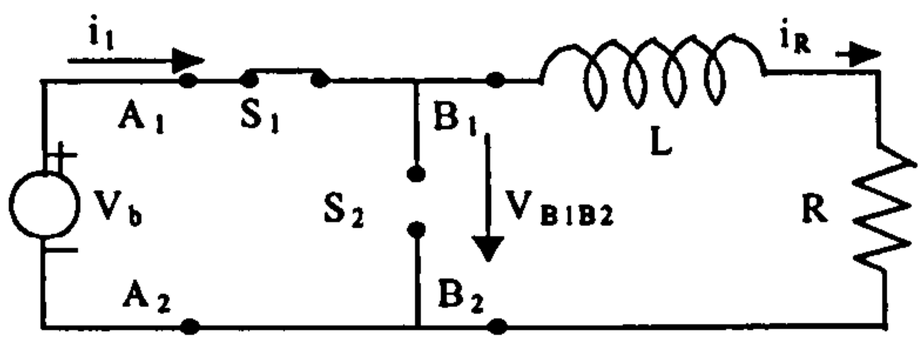
Figure 5.13. Chopper dc-dc conversion.

The input terminals of the chopper are A_1 (positive) and A_2 (negative). The output terminals are B_1 (positive) and B_2 (negative). The load resistance R is connected to the output through an inductance L. The purpose of this inductance is to smooth out fluctuations in the output current caused by the switching process in the chopper. If L is sufficiently large, the voltage across the load and the current through it will be substantially dc, with negligible ac ripple. The inductance L, which performs the function of filtering the ac components, is usually iron-cored and should be capable of carrying the full dc load current without magnetic saturation.

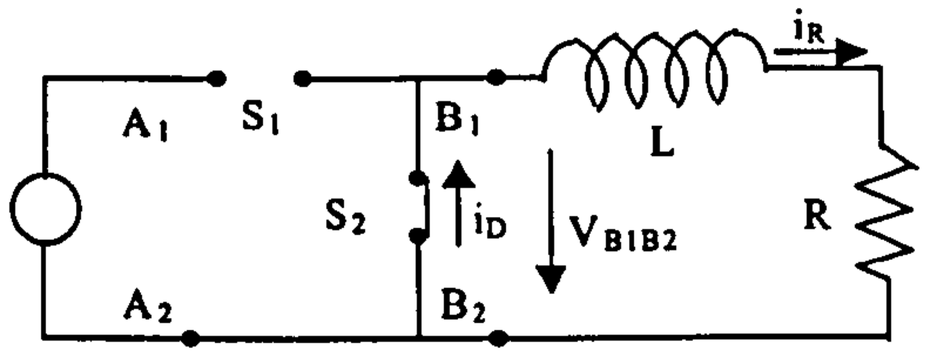
In addition to the power circuit shown in the figure, the chopper has a control circuit. The control circuit provides the switching signals to operate the power semiconductor static switch S_1 to turn it on or off as required. To derive quantitative relationships, the following definitions are made as:

f_s	repetitive switching frequency of the chopper (Hz)
T_s	cycle time of the chopper=$1/f_s$
T_{on}	on time of switch S_1
T_{off}	off time of switch S_1
D	duty cycle of switch S_1, defined as D= T_{on}/T_s.

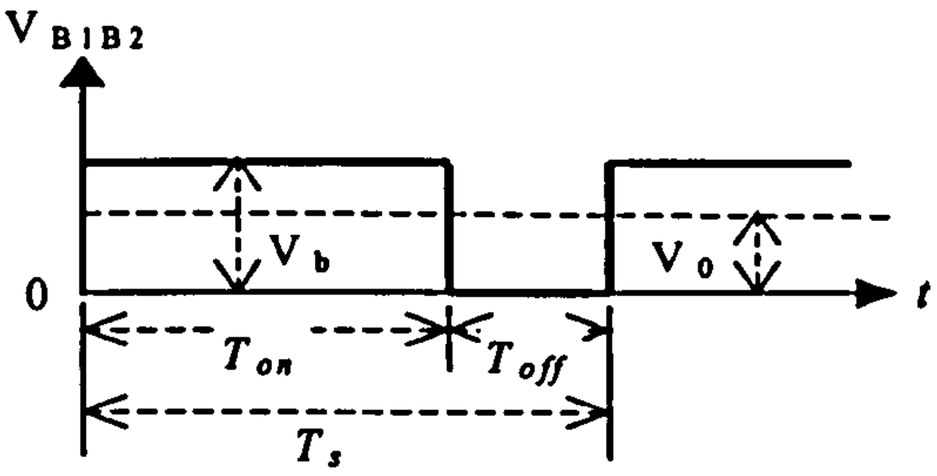
In Figure 5.13, when S_1 is turned on, the voltage V_1 is applied in reverse across the power diode. Therefore, the power diode, which is labelled as static switch S_2, must stay off as long as S_1 remains on. The switch S_1 is kept on for a time interval T_{on}, and then turned off. At the instant when S_1 is turned off, i_R has a finite value. It is the peak value of the output current during this chopper cycle. This current cannot instantly fall to zero, because of the presence of the inductance L. The decay of i_R causes an induced voltage Ldi_R/dt across the inductance. Because of this voltage, the diode S_2 gets forward-biased and carries the "freewheeling" current due to the stored energy in the inductor. Therefore, the turn off of S_1 automatically causes the turn on of S_2 when an inductance with stored energy is present. Therefore, depending whether S_1 is on or off, the converter operation can be described with two circuits as shown in figures 5.14 (a) and (b). The consequence of S_1 being on is the build up of current in the load resistance. The current i_R in the load circuit builds up exponentially from an initial zero value at the starting of the dc-dc converter because of the inductance L. The waveform of the voltage across the output terminals B_1- B_2 is equal to V_b when S_1 is on as shown in Figure 5.14 (c).



(a)



(b)



(c)

Figure 5.14. Equivalent circuits of chopper operation

(a) on state of S_1 (b) off state of S_1 (c) waveform of V_o

When S_1 is not conducting during T_{off}, the freewheeling diode current, labelled as i_D, is the load current i_R which decays exponentially. The decay of i_R continues before it reaches zero, as long as S_1 is off, that is for duration T_{off}. During T_{on}, the output current i_R is the same as the input current i_1.

The next chopper switching cycle commences when S_1 is turned on again. The circuit configuration again changes to that of Figure 5.14 (a) and the current again starts to build up. In this way, the circuit will reach the stable repetitive state as the switching progresses. This means that the peak current is effectively the same as that in successive cycles. Notice that the waveform of the voltage across B_1- B_2 is repetitive from the first cycle itself. As far as the energy conversion is concerned, during the interval when the switch S_1 is on, the input provides energy to the load as well as to the inductor. During the interval when the switch S_1 is off, the inductor current flows through the diode S_2, transferring some of its stored energy to the load.

5.2.3. Voltage relationship between input and output

Figure 5.14 (c) shows the waveform of the voltage V_{B1B2} at the output terminals B_1 and B_2 of the chopper. This is a train of rectangular pulses with width of T_{on}. This voltage consists of a dc component and an ac component. The ac component is the "ripple voltage". The inductance L has the function of absorbing the ripple voltage across it, so that only the dc component is present to the load resistor R. Notice that the ripple frequency is the same as the chopper switching frequency. With a sufficiently large inductance and a high switching frequency, the ripple component of the output voltage can be made negligibly small. The average output voltage can be calculated in terms of the switch duty ratio:

$$V_o = \frac{1}{T_s}\int_0^{T_s} V_{B1B2}(t)dt = \frac{1}{T_s}\left(\int_0^{on} V_d(t)dt + \int_{on}^{T_s} 0\cdot dt\right) = \frac{t_{on}}{T_s}V_d = DV_d \qquad (5.14)$$

where D is the switching duty cycle of the chopper, defined as the ratio of on time to total cycle time.

As described, the chopper functions as an ideal voltage step down transformer in dc systems, similar to a step-down transformer in ac systems. In the continuous current conduction mode, the corresponding relationships between input and output quantities are:

$$\text{Voltage ratio } V_o / V_d = D \tag{5.15}$$

Assuming a power lossless system:

$V_d I_d$ (input power) = $V_o I_o$(output power)

Therefore:

$$\text{Current ratio } I_o / I_d = 1/D$$

Theoretically, the voltage ratio of the dc transformer is adjustable in the range from zero to one through the simple means of adjusting the duty cycle D of the chopper. Ideally, with such a dc-dc converter, any output voltage lower than the input voltage can be obtained, even if the source voltage changes in a certain range.

5.2.4 Chopper circuit with filters

In the chopper circuit illustrated in Figure 5.14 (c), the output voltage varies between V_d and zero, which is normally not suitable to practical applications. The output filter, as shown in Figure 5.15, may be added into the circuit to reduce the voltage fluctuations. The output voltage fluctuations are very much diminished by using a low-pass filter, consisting of an inductor and a capacitor. The corner frequency f_C of this low-pass filter is selected to be much lower than the switching frequency, thus essentially eliminating the switching frequency ripple in the output voltage.

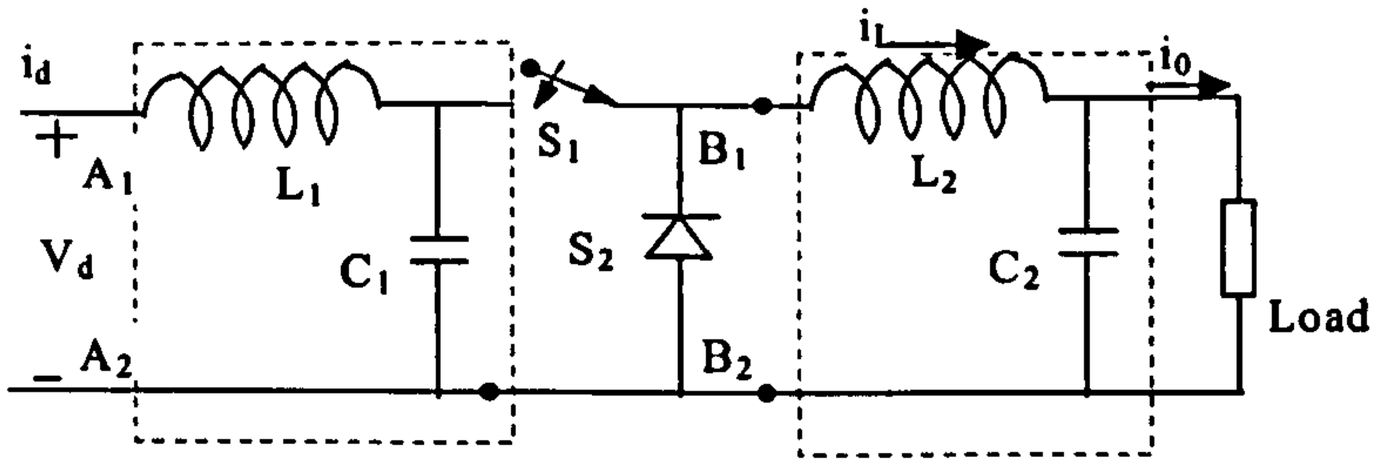


Figure 5.15. A chopper circuits with input and output filters

At the input side, an input filter shown in Figure 5.15 can also be used to smooth the input current from the rectifier bridge.

For the steady-state analysis, with respect to the chopper circuit shown in figure 5.15, it may be assumed that the filter capacitor at the output is relatively large, consequently the load circuit time constant would be large, so that the load current could be considered as ripple-free. In a step-down converter, the average inductor current I_L is equal to the average output current I_0, therefore the capacitor carries ripple current only.

The voltages and currents of interest include the currents through the switching devices and the voltages across them, which determine the ratings of the devices. However, a safety factor in the voltage rating is necessary, in order to take care of unavoidable overvoltage spikes, which may occur during switching transients, also a safety factor in the current rating should be considered to take care of overcurrents due to abnormal operating conditions. Ideally, a small on-state voltage drop and short switching time of the device are preferred. All the above are mainly based on the following considerations:

- The on-state voltage or on-state resistance dictates the conduction losses in the device;
- The switching time dictates the energy loss per transition and affects the choice of the operating frequency;
- The ratings of voltage and current determine the device power-handling capability, and the device cost.

5.2.5. Chopper control

The pulse-width modulation (PWM) switching method can be used for controlling the output voltage. Generally speaking, both the switching frequency and the on duration of the switch signal can be varied. However, the variation of the switching frequency makes it difficult to filter the ripple components. Therefore, switching at a constant frequency is more preferable. In this application, constant frequency switching (hence, a constant switching time period $T_s = t_{on} + t_{off}$) is employed and adjusting the

switching duty cycle D is used to control the average output voltage. As shown in figure 5.16, the switch control signal has a constant frequency.

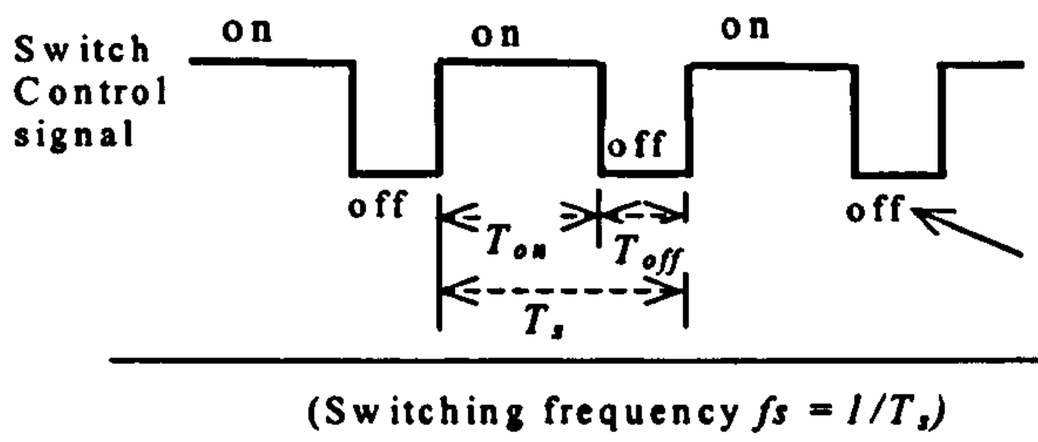


Figure 5.16. Switching control scheme of a chopper circuits

From equation (5.15), it can be seen that the switching duty cycle, D, has to be changed if V_d varies while V_o is kept constant. In normal operation, the system load changes from time to time, consequently, the machine speed and internal voltage drop in the system will vary. Therefore, the dc-dc converter input voltage, V_d, will fluctuate.

A controller has been designed to perform adjustments of the switching duty cycle to keep the output voltage. The controller produces the required adjustment based on the deviation of the real voltage from the reference voltage. Figure 5.17 pictures the block diagrams of the controller together with the simulation model of the dc/dc conversion system.

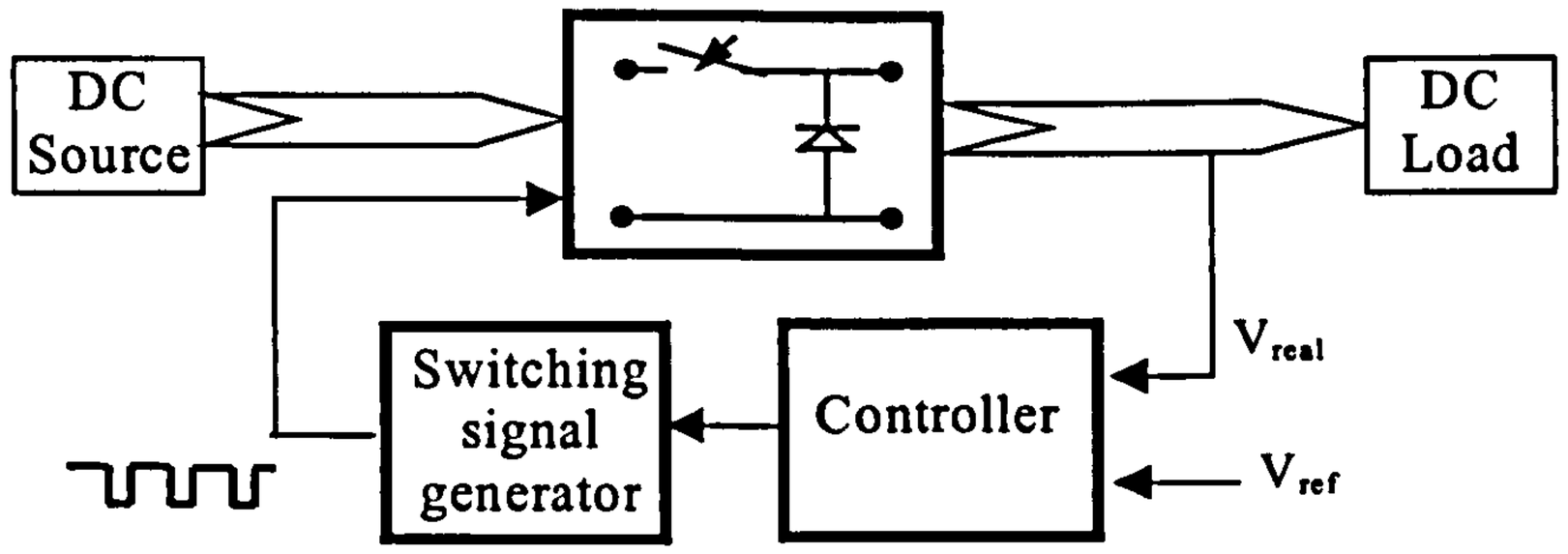


Figure 5.17. Block diagram of dc/dc conversion system

5.2.6. DC-DC converter simulation

Simulation models for the dc-dc conversion system and control system have been developed and simulation studies have been performed. Figure 5.18 shows the simulation results of a case study, where the load current is doubled at 0.03 seconds as shown in figure 5.18 (a). In Figure 5.18 (b), it can be seen that the input current increases correspondingly to supply the load requirement. It is also noted that the input current contains rich current pulses due to the chopper switching. Figure 5.18 (c) shows the output voltage drops when the change of the load current occurs, however, the control system adjusts the ratio of switching duty cycle to bring the output voltage back to the reference level, which is 100V. The chopper input voltage is assumed as a constant level of 200V during the period of simulation, as shown in Figure 5.18 (d). The ripples in the input voltage are caused by the chopper switching. A high degree of ripple appears when a larger current is delivered after the change of the load current.

The results of the presented case study demonstrated the correctness of the developed simulation models for the power electronic conversion system and the associated control system. Further developments, including integrating the developed system into the overall system, can therefore be conducted on the basis of the developed system and strategies. These are discussed in the following chapters.

Before presenting the system control strategy in chapter 7, in the next chapter the fuzzy control method is discussed and the application in the variable speed control is described.

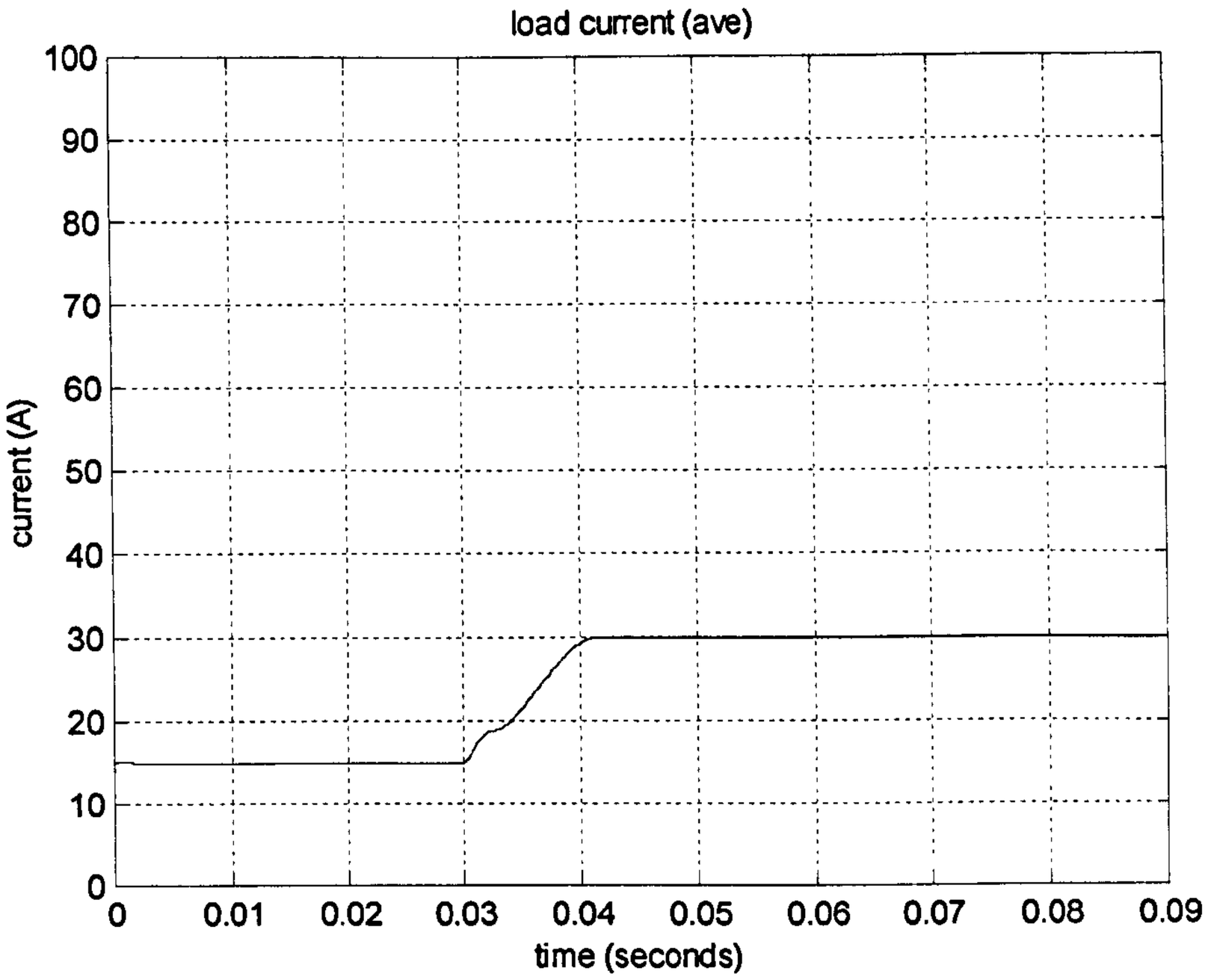


Figure 5.18 (a) Load current (avarage value)

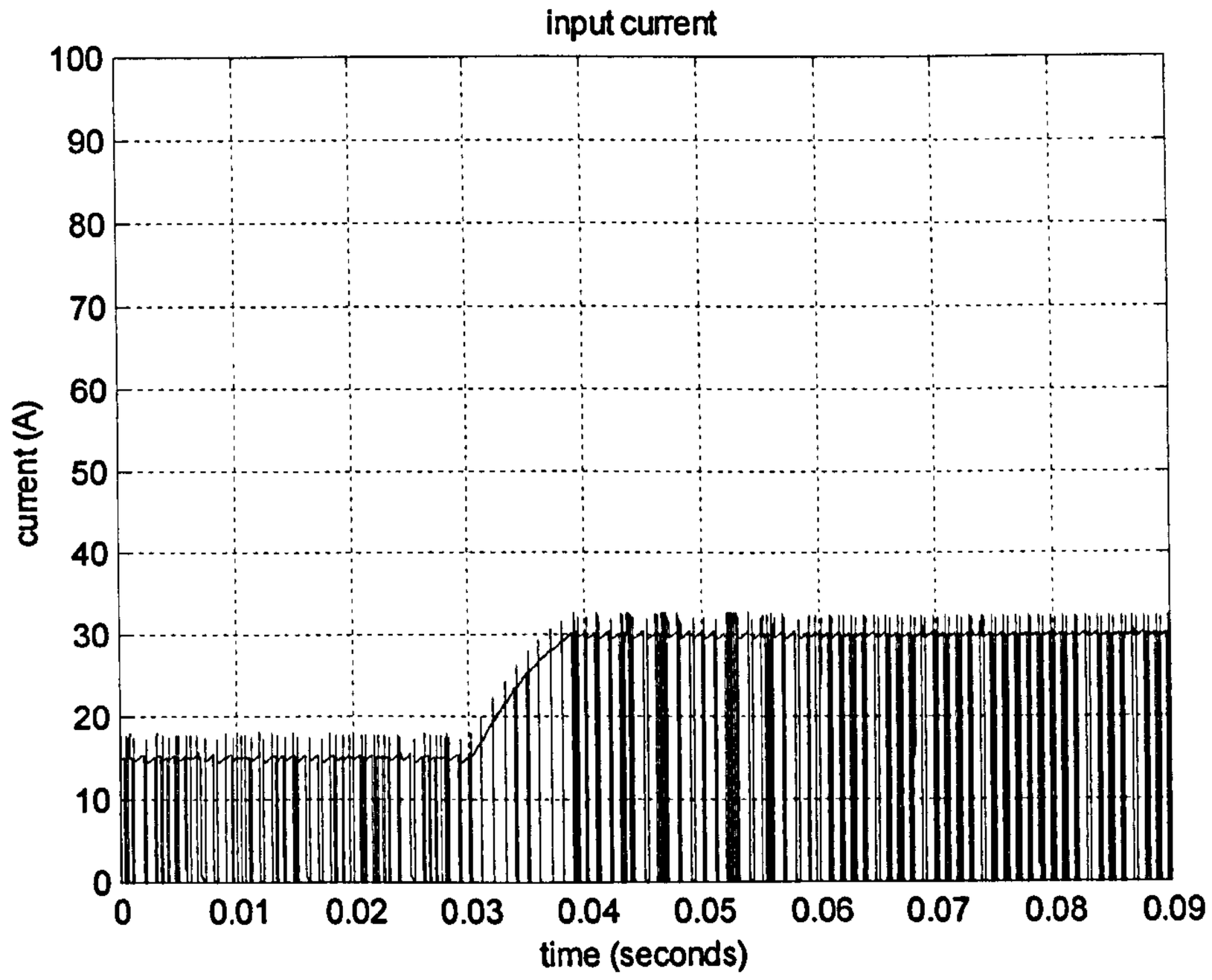


Figure 5.18 (b) Input current

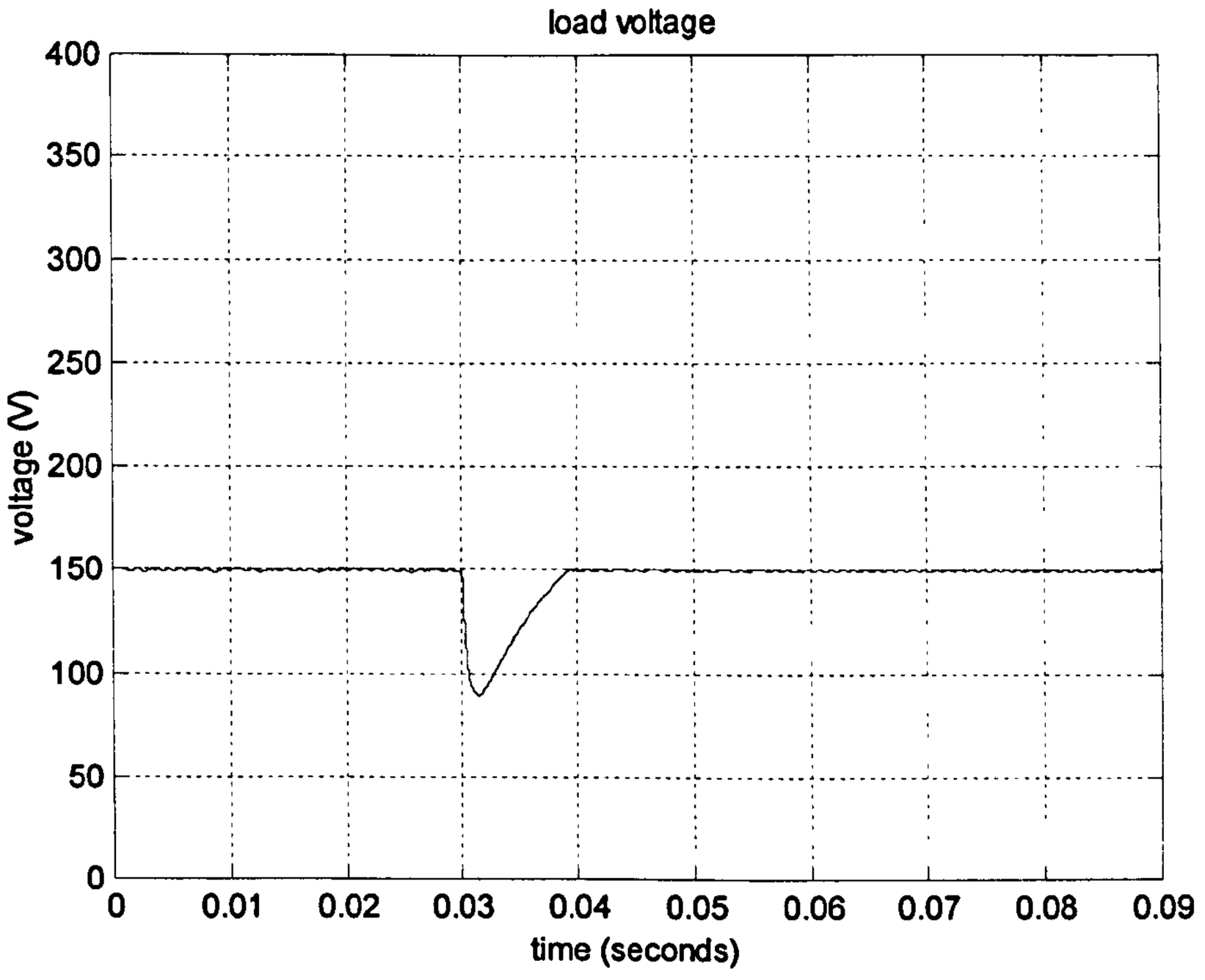


Figure 5.18 (c) Output voltage

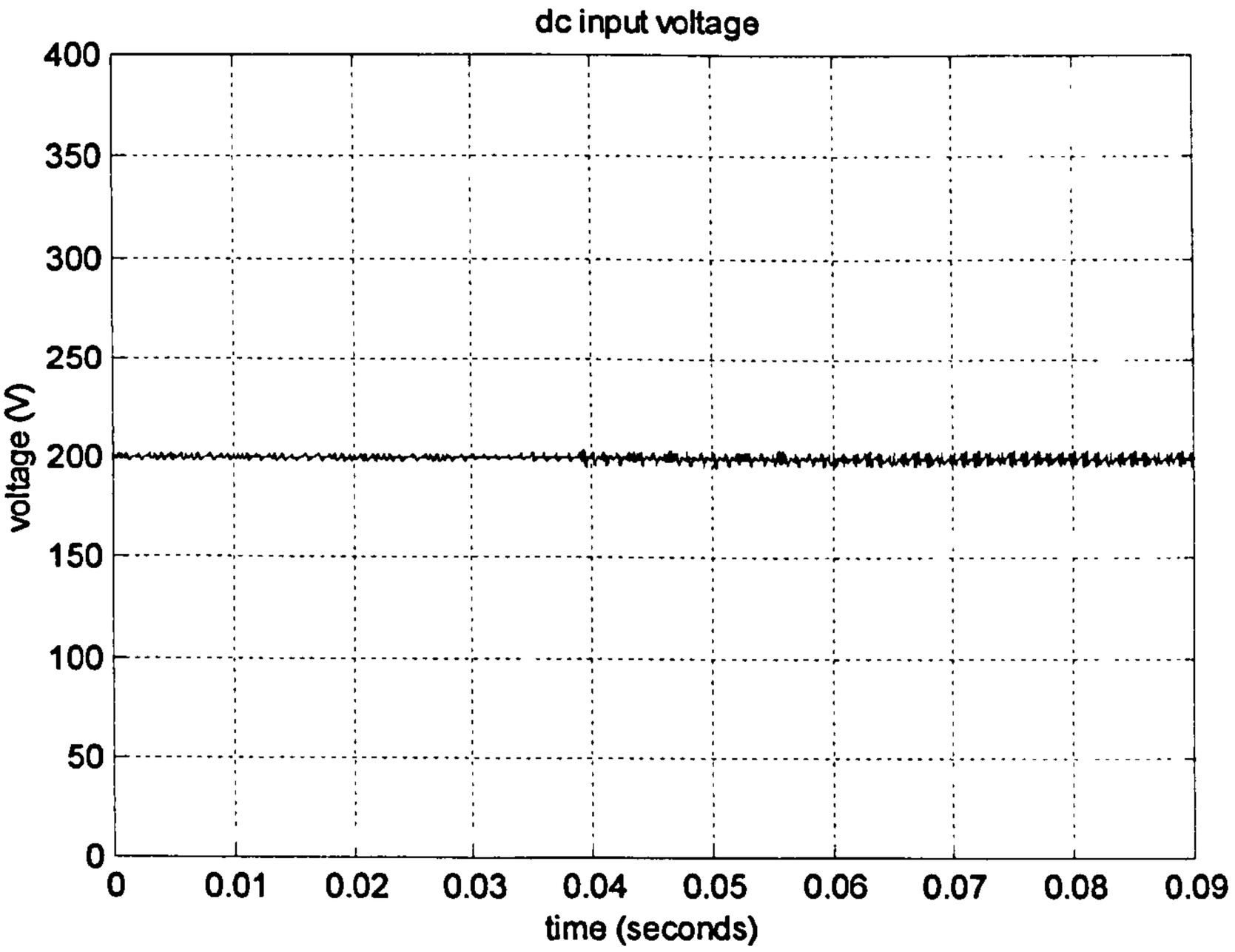


Figure 5.18 (d) Input voltage

Figure 5.18. Simulation results of dc-dc converter case study

Chapter 6

Hybrid Variable Speed Controller

In recent years, fuzzy logic has emerged in various applications. Fuzzy logic based control techniques typically have the following advantages, they are:

- conceptually easy to understand.
- tolerant to imprecise data.
- can deal flexibly with non-linear system because an accurate mathematical model is not required.

Fuzzy sets (in particular fuzzy numbers) and fuzzy logic applied to control problems form a field of knowledge called fuzzy logic control (FLC). The advantages make fuzzy logic suitable for control problems in an environment of uncertainty and imprecision. Classical control methodologies are usually based on precise mathematical models of the objects to be controlled. Fuzzy logic control methods can be combined with conventional control methods. Fuzzy logic controllers (FLCs) have already gained their reputations as robust and relatively low in computation requirements. In this thesis the diesel engine variable speed controller is designed based on a hybrid method, a fuzzy logic core is developed and embedded in the variable speed controller while a conventional control method is used for the outer loop of the controller.

6.1. Fuzzy logic controller structure

In conventional control, the control decision is determined by a number of input data, processed by a set of equations (the mathematics model) which represents the entire control plant. To express human experience in the form of mathematical formulae is a very difficult task. Fuzzy logic provides a simple tool to interpret experience into reality. Fuzzy logic controllers are rule-based controllers, which utilize the principles

of fuzzy set theory, their data representation and their logic relations. The basic configuration of a fuzzy logic controller can be simply represented by the four parts colored in gray background in Figure 6.1.

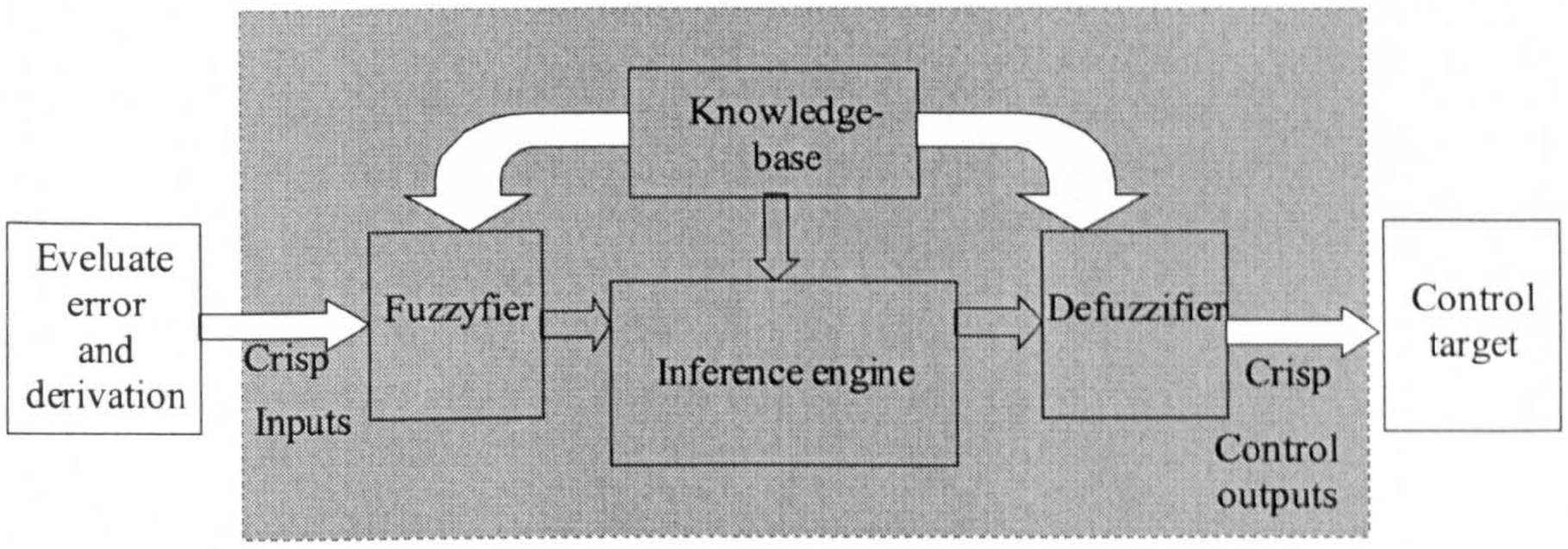


Figure 6.1. Basic structure of a fuzzy control system

Each part has a different function:

1. **Fuzzifier block** reads, measures and scales the controller input variables and transforms the measured numerical values into the corresponding linguistic (fuzzy) variables with appropriate membership values.
2. **Knowledge base block** includes the definitions of the fuzzy membership functions defined for each input and output variable and the rules that specify the control goals with linguistic variables.
3. **Inference block** simulates human decision-making and determines the control actions based on fuzzy logic.
4. **Defuzzifier block** converts the inferred decision from linguistic variables back to numerical values.

Such controllers can be based on a general scheme where an error and its derivative are evaluated and fuzzified by using suitable membership functions. The fuzzified signals are then sent to the inference block where control signals are deduced by evaluating with linguistic rules. Finally, the variation of the control signal is obtained with defuzzification, which can be achieved by means of various methods, for example, the centroid algorithm, which is used for the fuzzy logic core design in the project.

6.2. Foundations of the fuzzy sets theory

Traditional control techniques are to some extent limited by the need to make a compromise between the best correspondence of the mathematical model to the actual plant and the complexity of the related algorithm. Sometime it could happen that inadequate control actions arise from poor knowledge of the physical phenomena and/or from an inaccurate modeling. It is easy to verify that a non-linear system needs more complex and/or a higher order model to represent the system. Fortunately, in the situation where quantitative relations are unavailable, qualitative information is in general available and this can be used in performing the control actions with a fuzzy logic method.

Starting from such a background, "fuzzy sets" theory has been conceived as a technique being able to correctly manipulate and qualitatively analyse information with an inductive action similar to the human approach.

Before describing the design process, some terms used in fuzzy logic are introduced.

Membership functions

A membership function can be a curve that defines how each point in the input space is mapped to a membership value (degree of membership) between 0 and 1. The input space is sometimes referred to as the “universe of discourse”. The degree of an object belong to a fuzzy set is denoted by a membership value between 0 and 1. A membership function associated with a given fuzzy set maps an input value to its appropriate membership value. There are different shapes of membership functions. The simplest membership functions can be constructed using straight lines, for example, the triangular membership function.

Fuzzy sets

A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. To understand the fuzzy set, firstly we might consider a classical set. A classical set can be viewed as a container that wholly

includes or wholly excludes any given element. This means that an element x must be either set A or not-set A. Conversely, in fuzzy logic, fuzzy sets describe vague concepts. Any statement can be fuzzy and the truth of any statement becomes a matter of degree. The tool of fuzzy reasoning gives the ability to reply to a yes or no question with a not quite yes or no answer. Reasoning in fuzzy logic is just a matter of generalizing the familiar yes or no logic, which means if we give 'true' the numerical value of '1' and 'false' the numerical value '0', the value in between, for instance 0.4, is permitted and has meaning.

Using the mathematical expression, for a classical set might be expressed as:

$$\Re(x) = \{1 \mid x > m\} \qquad \text{or}$$

$$\overline{\Re}(x) = \{0 \mid x \le m\} \qquad \text{m can be a figure, for example m=100.}$$

A fuzzy set is an extension of a classical set. A fuzzy set admits the possibility of partial membership. If X is the universe of discourse and its elements are denoted by x, then a fuzzy set A in X is defined as a set of ordered pairs:

$$A = \{x\ ,\ \mu(x) \mid x \in X\},$$

$\mu(x)$ is the degree of membership in the fuzzy set A, called the membership function of x in A. The membership function maps each element of X to a membership value between 0 and 1.

Logic operators

Fuzzy logical reasoning is in fact a superset of standard Boolean logic. In other words, if we keep the fuzzy values at their extremes of 1 (completely true) and 0 (completely false), standard logical operations will hold. As an example, consider the standard Boolean logic operators in Table 6.1 and 6.2:

Table 6.1. Standard Boolean logic (AND, OR) truth table

A	B	A AND B	A OR B
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

and

Table 6.2. Standard Boolean logic (NOT) truth table

A	NOT A
1	0
0	1

In fuzzy logic the truth of any statement is a matter of degree, and the input values can be real numbers between 0 and 1. To preserve the results of the AND truth table and also to extend the logic operation of AND to any real number between 0 and 1, the MIN operation is applied to fuzzy logic. That is, that the statement A AND B is resolved, where A and B are limited to the range (0,1), by using the function MIN(A, B). Similarly, we can replace the OR operation with the MAX operation function, so that A OR B becomes equivalent to the operation MAX(A,B). The operation of NOT(A) becomes 1-A, therefore, values other than 1 and 0 may be considered. With these substitutions the standard Boolean-logic truth tables become the fuzzy logic operations shown in Table 6.3 and 6.4. It can be seen that the relations for extreme fuzzy values, 0 and 1, are completely unchanged.

Table 6.3. Fuzzy logic operators (MIN, MAX)

A	B	MIN(A,B)	MAX(A,B)
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

And

Table 6.3. Fuzzy logic operator (1-A)

A	1-A
1	0
0	1

If-then rules

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic. The if-then rule statements formulate the conditional statements in fuzzy logic. A single fuzzy if-then rule assumes the form

If x is A then y is B

Where A and B are linguistic values defined by fuzzy sets (universes of discourse) X and Y, respectively. The if-part of the rule, "x is A", is called the antecedent or premise, while the then-part of the rule, "y is B", is called the consequent or conclusion. So the antecedent is an interpretation that returns a single number between 0 and 1. On the other hand, the consequent is an assignment that assigns the linguistic value B to the output variable y.

In general the input to an if-then rule is the current value of the input variable and the output is an entire fuzzy set, which will be defuzzified, assigning one value to the output.

Interpreting an if-then rule is a three part process.

1. Evaluate antecedent. Resolve all fuzzy statements in the antecedent to a degree of membership between 0 and 1.
2. Apply fuzzy operator to multiple part antecedents. If there are multiple parts to the antecedent, the fuzzy logic operators are applied to resolve the antecedent to a single number between 0 and 1.
3. Apply result to consequent. The degree of support for the entire rule is used to shape the output fuzzy set. The consequent specifies that fuzzy set be assigned to the output. If the antecedent is only partially true, i.e. being a value less than 1, then the output fuzzy set is truncated according to the implication method.

In general, a fuzzy logic controller has a number of rules. The output of each rule is a fuzzy set. The output fuzzy sets representing those rules are then aggregated into a single output fuzzy set. Finally the resulting set is defuzzified, or resolved to a single number.

Fuzzy inference

Fuzzy inference is the process of formulating the mapping from given inputs to outputs using fuzzy logic. The mapping then provides a basis from which decisions can be made. The process of fuzzy inference involves all of the pieces that are described in the above: membership functions, fuzzy logic operators, and if-then rules.

Mamdani's fuzzy inference method is commonly used in fuzzy methodology, which was among the first control systems built using fuzzy set theory. It was proposed in 1975 by Ebrahim Mamdani as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from expierenced human operators [32]. Mamdani's effort was based on Lotfi Zadeh's 1973 paper on fuzzy algorithms for complex systems and decision processes [51].

Mamdani's fuzzy inference method expects the output membership functions to be fuzzy sets. After the aggregation process, there is a fuzzy set for each output variable that needs deffuzification. In many cases, it may be much more efficient to use a singleton output membership function rather than a distributed fuzzy set. It enhances the efficiency of the defuzzification process because it greatly simplifies the computation by using the weighted average to find the centroid of a two-dimensional function rather than by integrating across the two-dimensional function.

6.3. Structure of the proposed hybrid variable speed controller

In the studied system, the generation of the required power relies on the fuel input to the prime mover. The principal task of the controller is to control the fuel input, therefore the controller output is the control signal to the actuator. If the dc link voltage is controlled close to the reference value, the output power is proportional to the dc link current, i.e. $P \propto I$, and hence the input fuel is directly related to the dc link current in the operation range. Therefore the current, which represents the required system power level, is used as an input parameter to a conventional controller to produce a reference speed. Considering the non-ideal features of the system, such as non-linearity and power loss, using the current as the only control parameter may not be sufficient, so the engine speed is also used as an input to the controller. This means that adjustment of the system operation while system load changes is decided by the dc link current and the speed of the generator.

The configuration of the variable speed controller consists of two loops. The outer loop decides the speed reference level that varies with load, while the inner loop produces the control signal based on the speed error and the derivative of the speed error with respect to the new speed reference. Figure 6.2 shows the configuration of the hybrid controller, where there is a fuzzy logic core located in a frame.

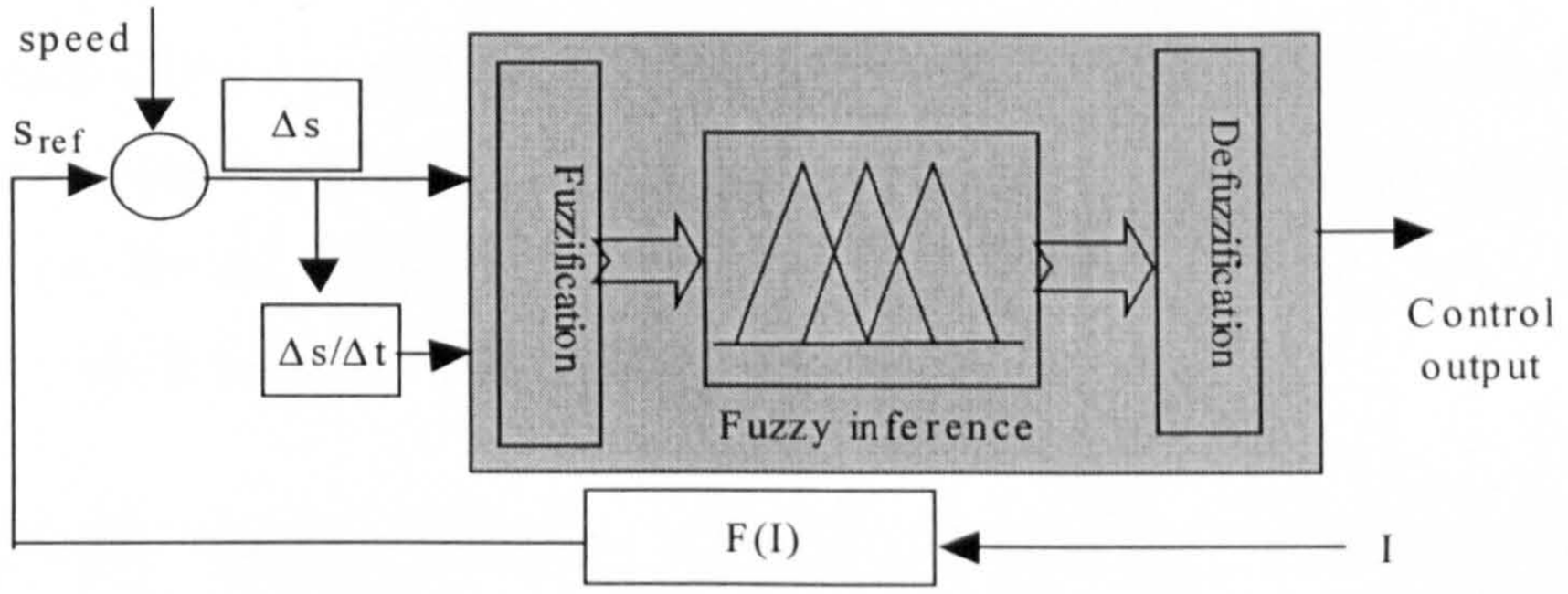


Figure 6.2. Schematic diagram of the hybrid variable speed controller

With the controller structure described Figure 6.2, the fuzzy logic core based on the fuzzy logic algorithm has been developed. Since the purpose of the fuzzy logic core is to generate the adjustment on the fuel input, ΔU, to control the speed of the engine generator system and hence to control the output power, the speed error and its derivative are selected as the fuzzy core inputs. The output is ΔU to the fuel control signal. Mamdani's direct method is used to materialize fuzzy reasoning and the defuzzification process is based on the center-of-gravity method. The design of the fuzzy logic core is discussed in the following sections.

6.4. Fuzzy logic core design

A fuzzy logic core has been designed for the proposed variable speed controller. The major steps of the design are described in this section.

6.4.1. Choice of control variables

The inputs and outputs of the controller depend on the applied control strategy. It is common to use the instantaneous error of the controlled variable with respect to its reference and the derivative of the error as controller inputs. Two such inputs are

chosen for the fuzzy logic core, the first input is the generator speed error and the second input is the change rate of the speed error, which describes the rapidity and direction of the speed change.

6.4.2. Defining membership functions

The input and output parameters of the fuzzy logic core are linguistic variables, that can be modeled by fuzzy sets. The number of fuzzy sets for each fuzzy variable varies according to the application. Increasing the number of fuzzy sets requires a corresponding increase in the number of rules.

Membership functions represent a number of quantizied overlapping fuzzy sets (linguistic variables) for each of the input signals to the fuzzy logic core and output signals from the core. The membership functions assigned to each fuzzy set map the crisp values into fuzzy values. Various shapes of membership functions can be found in fuzzy logic applications [5][38], the most often seen are triangular, trapezoidal or bell shapes. The triangular shape shown in Figure 6.3 is used for the controller described in this chapter. The linguistic variables NB, NM, NS, Z, PS, PM, and PB, as shown in Figure 6.3, stand for Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium and Positive Big respectively.

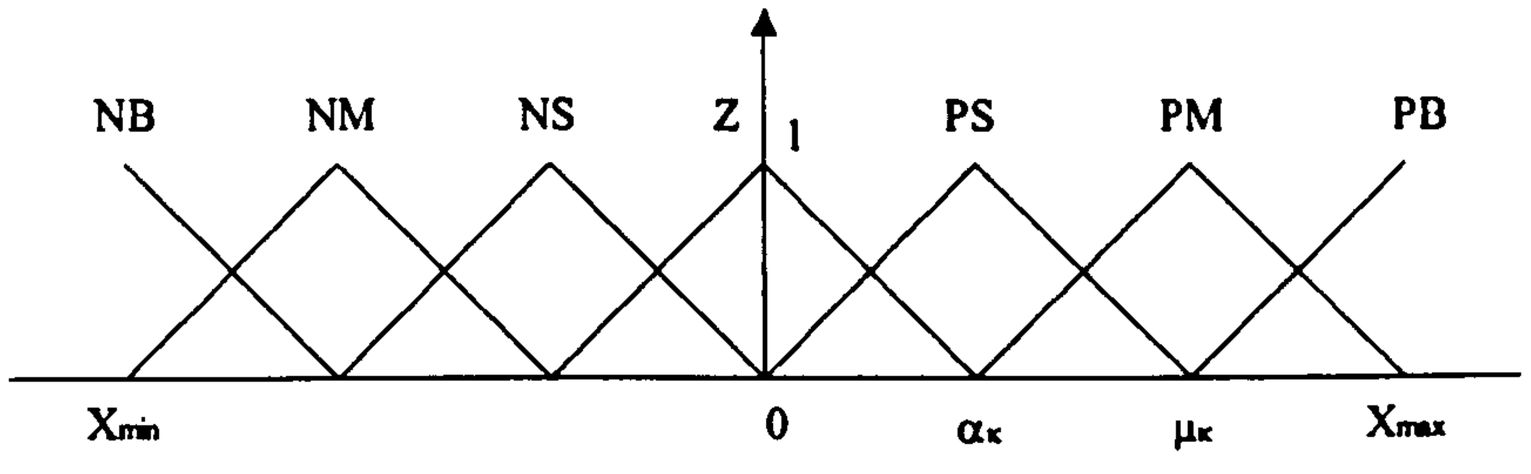


Figure 6.3. Seven triangular membership functions

The mathematical expression for the triangular shape is:

$$F_k(x) = \begin{cases} \dfrac{x - \mu_k + \alpha_k}{\alpha_k} & for(\mu_k - \alpha_k \le x \le \mu_k) \\ \dfrac{x - \mu_k - \alpha_k}{\alpha_k} & for(\mu_k \le x \le \mu_k + \alpha_k) \end{cases} \tag{6.1}$$

where k =1, ... n, n is the number of linguistic variables, μ_k is the centroid of the kth membership function and α_κ is a constant that determines the spread of the kth membership function.

In Figure 6.3, X_{min} and X_{max} are the minimum and maximum values of the control variable x. The membership functions are normalized in the range X_{min} and X_{max}, and symmetrical around zero. Making the membership functions symmetrical and each one overlaps with the adjacent functions by 50% can simplify computation [5].

6.4.3. Rule base and inference

In a conventional controller there are "control laws" to govern the operation of the controller. In a fuzzy logic controller, the equivalent term is "rule". Rules are linguistics in nature and allow the control decision to be made. In fuzzy systems, an input fuzzy set is mapped to an output fuzzy set. Fuzzy rules describe the relationship between the input and output fuzzy sets. They usually are in the form "if *A*, then *B*", where *A* is the rule antecedent and *B* is the rule consequence. The antecedents of each fuzzy rule describe a fuzzy input region in the state space. Once the fuzzy if-then rules are established, the control strategy can be realized by fuzzy reasoning. The different methods of fuzzy reasoning can be applied to make different types of fuzzy controllers.

Considering the fuzzy system with two antecedents (the error of speed, Err and the derivative of the error, DErr) and one consequence (the control signal U), each antecedence is quantized to seven fuzzy sets of Figure 6.3. Each pair of input fuzzy sets is mapped to one of the output fuzzy sets, then the relationship between the fuzzy input pairs and the output fuzzy set can be expressed as a rule matrix shown in Figure 6.4. Every element in the matrix represents a fuzzy rule. In another words, each of fuzzy rule maps two input fuzzy variables, the error, Err and the derivative of error, DErr, into one output fuzzy variable, i.e. the rule of "A_i and B_i, then C_i" maps inputs Err_i, $DErr_i$ to U_i. For example, if Err is negative big and DErr is positive big, which requires an opposite (negative) control action, then the fuzzy logic controller generates a negative big output to adjust the system. In the case where the error and error

derivative are positive, the implication is that the system error is decreasing and thus the magnitude of the control action may be kept unchanged. If the error becomes positive and the error derivative goes negative, then there is the necessity of an opposite (positive) control action. These rules can be expressed as:

If Err is NB and DErr is PB, then $U_{NB,PB}$ is NB;

If Err is PS and DErr is PS, then $U_{PS,PS}$ is Z;

… …

If Err is PB and DErr is NB, then $U_{PB,NB}$ is PB.

DErr

Err	NB	NM	NS	Z	PS	PM	PB
NB	Z	NS	NS	NM	NM	NB	NB
NM	PS	Z	NS	NS	NM	NM	NB
NS	PS	PS	Z	NS	NS	NM	NM
Z	PM	PS	PS	Z	NS	NS	NM
PS	PM	PM	PS	PS	Z	NS	NS
PM	PB	PM	PM	PS	PS	Z	NS
PB	PB	PB	PM	PM	PS	PS	Z

Figure 6.4. Fuzzy control rule matrix

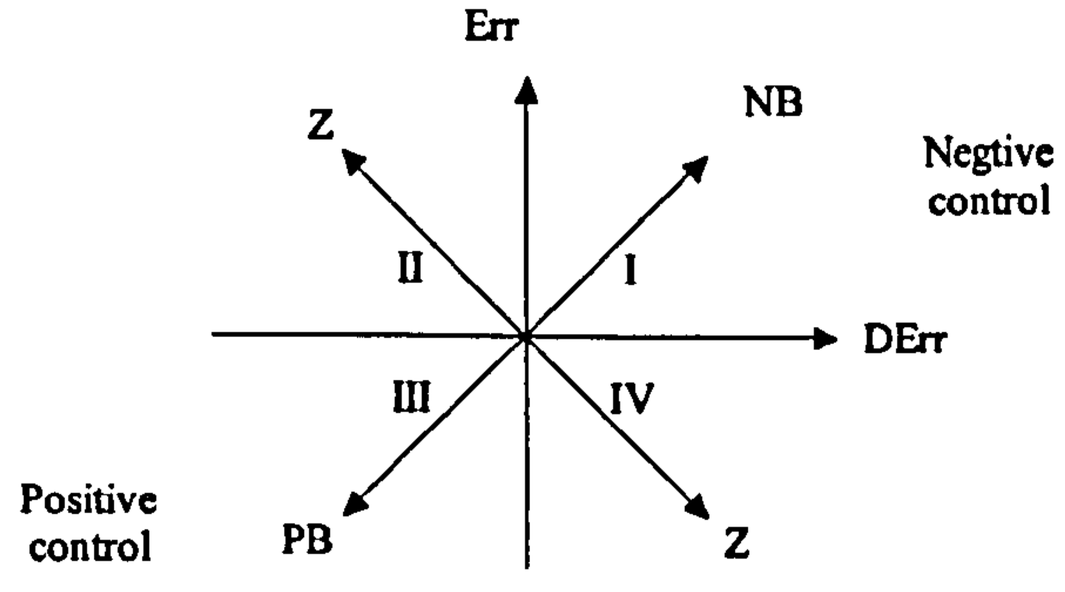


Figure 6.5. Phase plane for the controller fuzzy variables and control actions

The phase plane in Figure 6.5 can be seen as four sectors separated by two lines. In each sector different control action will be taken. The z line separates the negative control from the positive control, which represents the boundary of the directions of control actions.

The corresponding output fuzzy set U is the combination of the partially activated sets $U_{i,j}$, i.e., $U = \sum_{i=1,j=1}^{m,m} U_{i,j}$, as illustrated in Figure 6.6, where (i,j) is the number of membership functions of each input pair. Here, $\sum$ represents the aggregation operation.

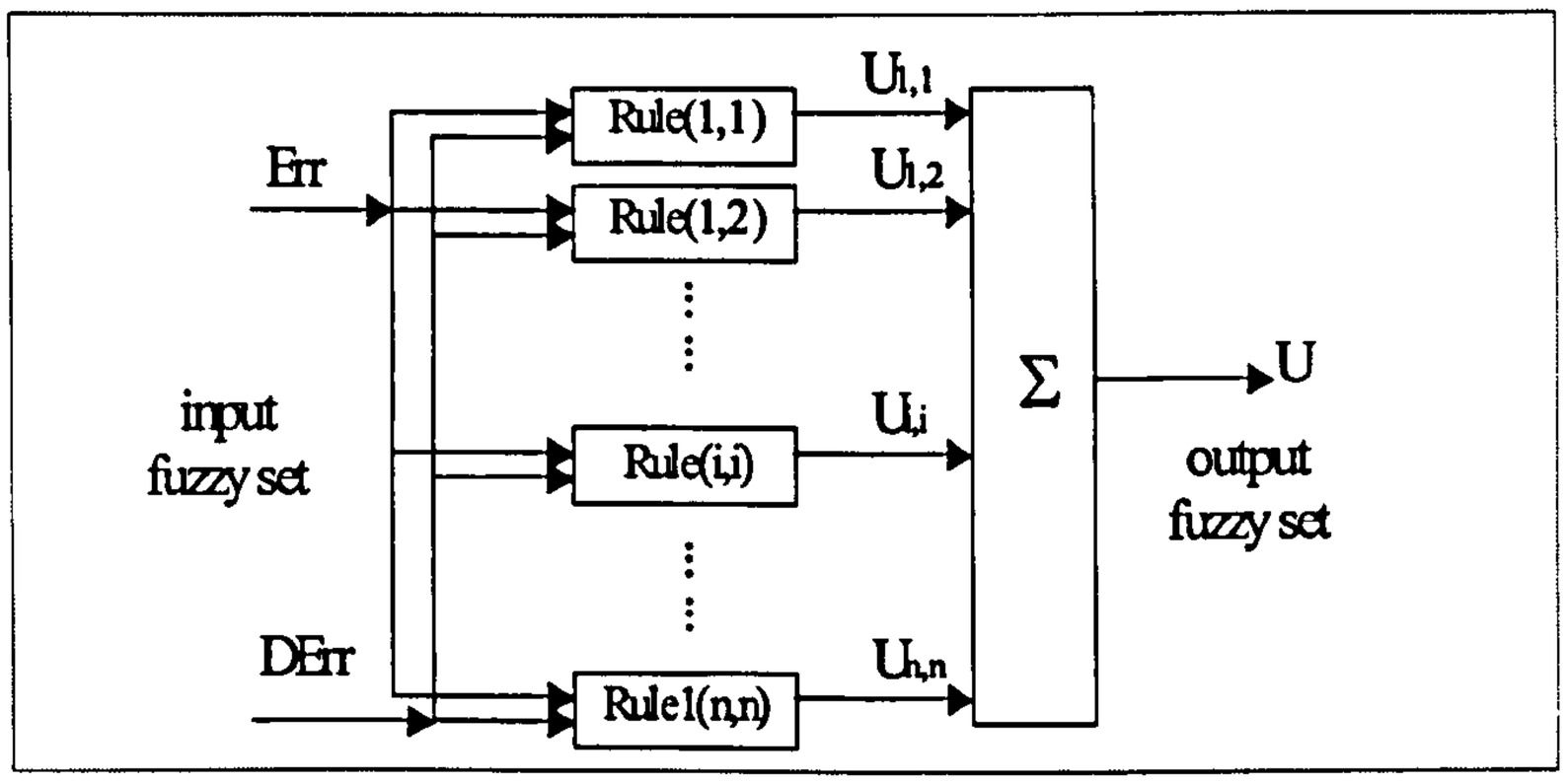


Figure 6.6. Fuzzy associative architecture

The activation of the (i,j)th rule consequent is a scalar value $U_{i,j}$, which equals the minimum of the two antecedent conjuncts' values with Mamdani's direct method. For example, if Err belongs to NB with a membership of 0.3 and DErr belongs to NM with a membership of 0.7, then the rule consequence ($U_{i,j}$) is 0.3.

6.4.4. Fuzzy inference

The mechanism of Mamdani's direct method can be used to materialize fuzzy reasoning by fuzzy relations. For the fuzzy speed control application two variables are in the premise part and one variable in the consequence part, so that the two-input one-output IF-THEN rules are used in the fuzzy reasoning. If we take rule (i,i) and (i+1,i+1) as example:

Rule (i,i): If x is A_i and y is B_i then z is $U_{i,i}$

Rule (i+1,i+1): If x is A_{i+1} and y is B_{i+1} then z is $U_{i+1,i+1}$ (6.2)

where A_i,, A_{i+1}, B_i, B_{i+1}, $U_{i,i}$, and $U_{i+1,i+1}$ are fuzzy sets. The reasoning process is illustrated in Figure 6.7.

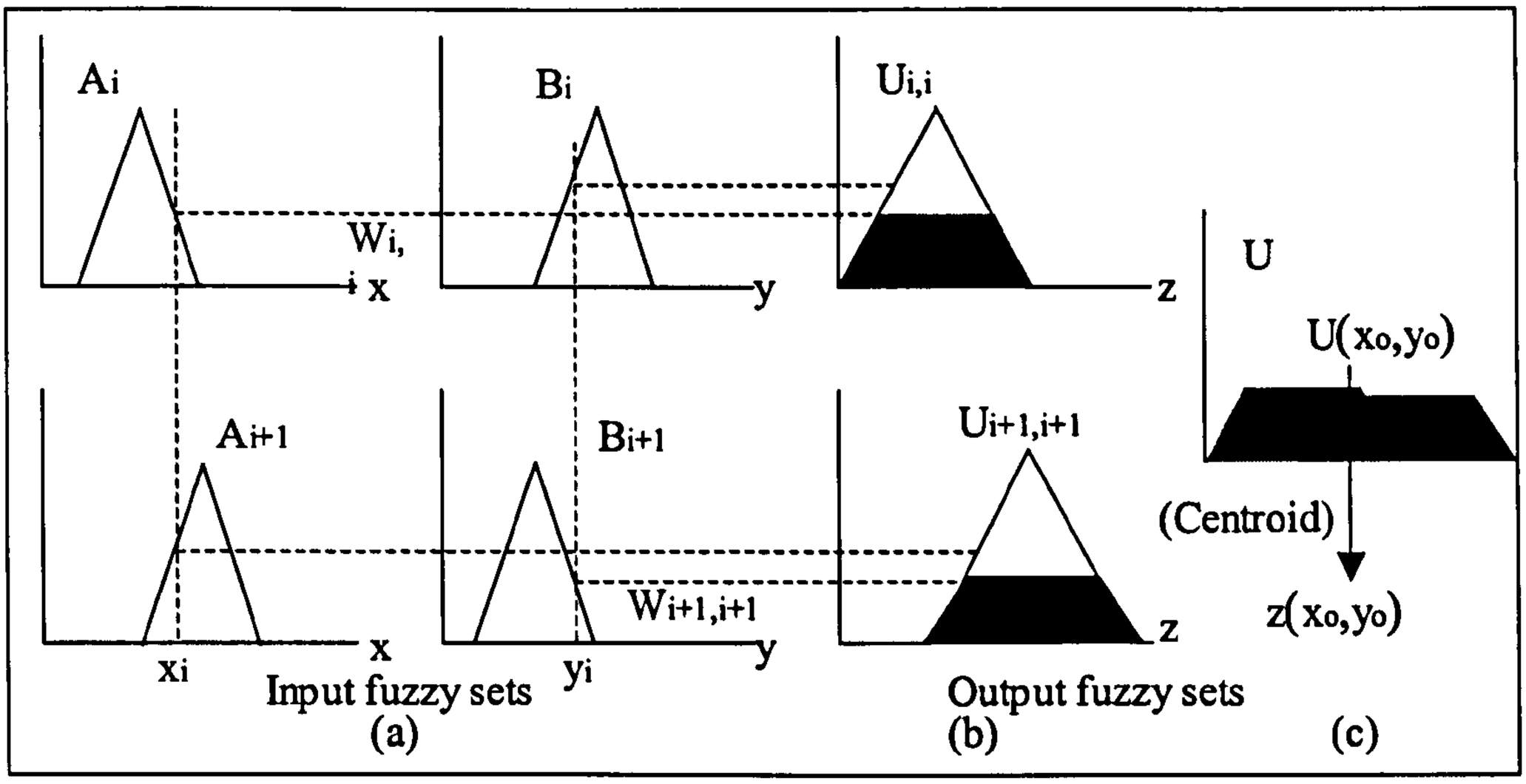


Figure 6.7. Reasoning process of Mamdani's direct method.

Suppose x_0 and y_0 to be the inputs for the premise part variables x and y respectively and denoted as (x_0, y_0), the reasoning process for the input pair (x_0, y_0) is

Step 1.

In the rules of (6.2), there are two variables x and y, in the premise part. Accordingly, we need to obtain two membership values for the two-input: x_0, y_0. If we apply MIN-operation, represented by ^, to the preceding membership values, we get the adaptability of each rule.

Adaptability of rule (i,i): $W_{i,i} = \mu A_i (x_0) \wedge \mu B_i (y_0)$

Adaptability of rule (i+1,i+1): $W_{i+1,i+1} = \mu A_{i+1} (x_0) \wedge \mu B_{i+1} (y_0)$ (6.3)

where μ represents the membership and W is the weighting factor. The above relationship is also illustrated in Figure 6.7(a).

Step 2.

Applying the MIN operation to the adaptability obtained in Step 1 and the fuzzy sets in the consequence part, the conclusion of each rule can be obtained as follows.

Conclusion of rule (i,i): $\mu U_{i,i} (x_0, y_0) = W_{i,i} \wedge \mu U_{i,i}$

Conclusion of rule (i+1,i+1): $\mu U_{i+1,i+1} (x_0, y_0) = W_{i+1,i+1} \wedge \mu U_{i+1,i+1}$

(6.4)

The calculation of (6.4) corresponds to cutting the fuzzy sets in the consequence part by the height of the adaptability of the premise in Figure 6.7(b), the results are the shaded areas.

Step 3.

The MAX operation is used to obtain the final conclusion by aggregating conclusions obtained from all of the rules in Step 2. If we apply MAX-operation, represented by $\vee$, to the preceding membership values, the final conclusion from the two inputs and one output fuzzy system is as follows.

Final conclusion is: $U(x_0, y_0) = \mu U_{i,i}(x_0, y_0) \vee \mu U_{i+1,i+1}(x_0, y_0)$ (6.5)

This gives the result of the reasoning process, which is illustrated in Figure 6.7(c).

Take a simple example, if the two rules are:

Rule (i,i): If x is NM and y is NS then $U_{i,i}$ is NS

Rule (i+1,i+1): If x is NS and y is NS then $U_{i+1,i+1}$ is Z

then the conclusion $U(x_0, y_0)$ will be the aggregation of NS and Z.

In general,

$$U(x_0, y_0) = \mu U_{1,1}(x_0, y_0) \ldots \vee \mu U_{i,i}(x_0, y_0) \ldots \vee \mu U_{n,n}(x_0, y_0) \qquad (6.6)$$

6.4.5. Defuzzification process

Defuzzification plays an important role in fuzzy logic based control system. It is the process in which fuzzy quantities defined over the output membership functions are mapped into a non-fuzzy (crisp) number. The input for the defuzzification process is a fuzzy set, which is the result from the aggregate process and the output of the defuccification is a single number.

Note that the final deductive conclusion discussed is given by a fuzzy set. This is not practical as a definite value is required as the output of controller. Therefore, the resultant fuzzy set needs to be converted to a definite value. In Figure 6.7 $z(x_0, y_0)$, the definite value, is obtained by defuzzification.

The defuzzification operation converts the inferred control actions of the fuzzy logic controller from fuzzy values to crisp values. The process depends on the output fuzzy set, which is generated from the activated rules. The output fuzzy set is formed by Mamdani's direct method, which is defined by the function U(x, y) shown in Figure 6.7. The general overall conclusion obtained by the aggregation is in the form of equation (6. 6). Rewriting the equation in a short form as:

$$U(x,y) = \sum (W_{i,j} \wedge \mu U_{i,j}) \tag{6.7}$$

where (i,j) are the subscripts of rules from (1,1) to (n,n) for the total number of membership functions of all fuzzy inputs, $\sum$ is the aggregating operant.

A deffuzification rule, the gravity center of the fuzzy set, is applied to obtain the control output $z(x_0, y_0)$, in general the rule can be expressed as:

$$z(x,y) = \frac{\sum (W_{i,j} \wedge \mu U_{i,j})}{\sum W_{i,j}} \tag{6.8}$$

Equation (6.8) is the mean of discrete output universe of discourse $\mu U_{i,j} \in z$, derived from the output fuzzy set, *z(x,y)* is a definite value.

6.4.6. Simplified rule evaluation

The measured values of input parameters have to be translated to proper terms of the corresponding linguistic variables. Normally a crisp value has to be matched against the appropriate membership functions representing the terms of the linguistic variable. The matching is defined as coding the inputs or fuzzification as illustrated in Figure 6.8.

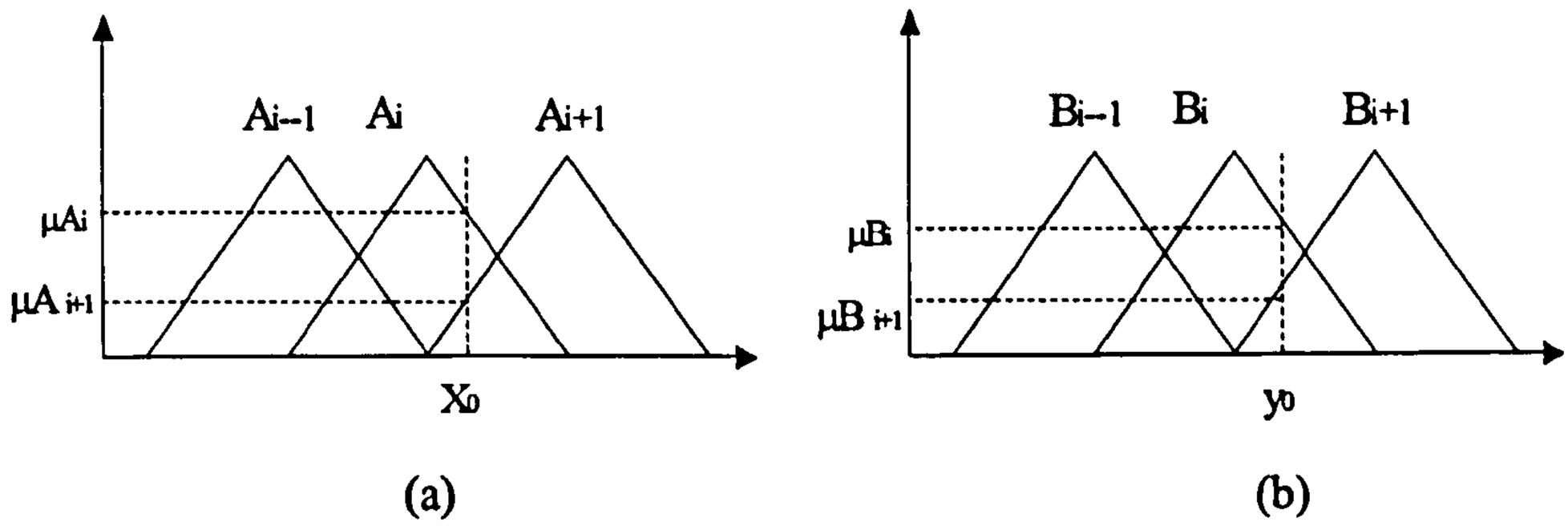


Figure 6.8. Fuzzy inputs corresponding to crisp input x_0 and y_0

In this two-input, one-output fuzzy logic control core design, symmetrical membership functions with 50% overlap are used, as shown in Figure 6.8, in which there are two values, μA_i and μA_{i+1} corresponding to x_0 in Figure 6.8(a). In the same way we can obtain the fuzzy inputs for y_0 in Figure 6.8(b), μB_i and μB_{i+1}, correspondingly.

Note that in both figures only membership functions A_i and A_{i+1}, B_i and B_{i+1}, intersect with x_0 and y_0 respectively thus producing the fuzzy values μA_i and μA_{i+1}, μB_i and μB_{i+1}, the others produce 0 value with membership functions equal to 0. This means that for every crisp input x_0 and y_0, only four rules are activated [5]. This fact results in a fuzzy decision table shown as Figure 6.9. Use of this feature will significantly reduce the calculation burden of the fuzzy logic controller.

	0	…	$\mu B_i(Y_0)$	$\mu B_{i+1}(Y_0)$	…	0
0	0	…				
⋮	⋮					
$\mu A_i(x_0)$		…	$U_{i,i}$	$U_{i,i+1}$	…	
$\mu A_{i+1}(x_0)$		…	$U_{i+1,i}$	$U_{i+1,i+1}$	…	
⋮	⋮					
0					…	0

Figure 6.9. Fuzzy decision table

Recall the defuzzification equation (6.8):

$$z(x,y) = \frac{\sum (W_{i,j} \wedge \mu U_{i,j})}{\sum W_{i,j}}$$

Due to a maximum of four rules are fired at a time, thus the above equation can be simplified. For the four fired rules as shown in Figure 6.9, the weighted-average consequence:

$$z(x_0, y_0) = m_{i,i}U_{i,i} + m_{i,i+1}U_{i,i+1} + m_{i+1,i}U_{i+1,i} + m_{i+1,i+1}U_{i+1,i+1} \quad (6.9)$$

$U_{i,i}$, $U_{i,i+1}$, $U_{i+1,i}$, $U_{i+1,i+1}$ are fired output sets, where weighting factors:

$$m_k = w_k / \sum w_k \qquad \text{k=(i,i), (i,i+1), (i+1,i), (i+1,i+1).} \quad (6.10)$$

Using the fuzzified input data set for Err and DErr, m_k can be calculated. Then the fuzzy control output can be generated using the simplified equations (6.9) and (6.10).

6.4.7. Parameter tuning

Selecting suitable parameters for the fuzzy logic controller plays an important role in achieving the goals of the controller. If sufficient information is not available about the controlled system, the selection of suitable parameters for the fuzzy logic controller can adopt a trial-and-error process, which may be quite time consuming. Efforts have been made to tune the fuzzy logic controller parameters at the design stage to obtain a simple solution and achieve an optimal or near optimal system performance.

A method used to tune the fuzzy logic controller parameters is off-line tuning. The objective of the parameter tuning method is to change the inputs and output membership threshold level and the reasoning rules in an organized manner to achieve desired system response. The tuning method tries to minimize three system performance indices: overshoot, steady state error and the transient time to reach a new steady state after a disturbance.

Parameter tuning can use the simulation results of a PI controller. Under the guidance of the simulated results, the trial-and-trace process changes the parameters in the fuzzy logic core. The adjusted parameters are then simulated to test system performance. This procedure can find whether one set of parameters meets the performance requirements or leads to instability. Alternatively in the case where system information is lacking to obtain the best parameters for the system may require a large number of iterations in searching for a set of desired parameters.

The following is a simple fuzzy logic controller model, the trial and trace process results are used for the controller fuzzy core test.

```
[System]
Name='fuzzy_controller'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
```

```
NumRules=9
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='du'
Range=[-60 60]
NumMFs=3
MF1='too_high':'trimf',[-60 -60 0]
MF2='good':'trimf',[-19.7 0 19.7]
MF3='too_low':'trimf',[0 60 60]

[Input2]
Name='du/dt'
Range=[-60 60]
NumMFs=3
MF1='slow_down':'trimf',[-60 -60 0]
MF2='steady':'trimf',[-20 -0.635 20]
MF3='speed_up':'trimf',[0 60 60]

[Output1]
Name='output1'
Range=[-1 1]
NumMFs=5
MF1='least':'trimf',[-1.468 -1 -0.48]
MF2='less':'trimf',[-1 -0.5 0]
MF3='average':'trimf',[-0.5 0 0.5]
MF4='more':'trimf',[0 0.5 1]
MF5='full_open':'trimf',[0.5 1 1.5]

[Rules]
1 0, 2 (1) : 1
2 0, 3 (1) : 1
3 0, 4 (1) : 1
1 1, 3 (1) : 1
2 2, 3 (1) : 1
3 3, 3 (1) : 1
1 3, 1 (1) : 1
3 1, 5 (1) : 1
2 2, 4 (1) : 1
```

6.5. Simulation of hybrid variable speed control system

Based on the fuzzy logic controller model discussed in the last section, the simulation model of the speed controller has been developed and simulation studies have been

performed to investigate the performance of the developed control system. Figure 6.10 shows the block diagram of the simulated system.

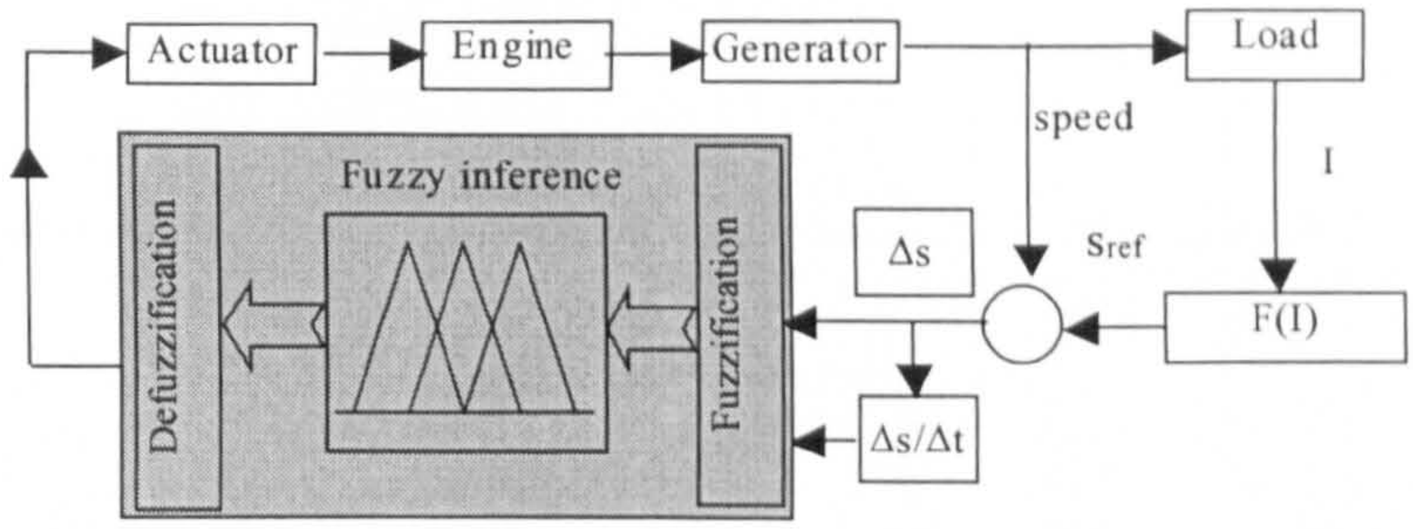


Figure 6.10. Block diagram of fuzzy logic core simulation study

Two case studies at varying load are presented in Figure 6.11 and 6.12. Figure 6.11 shows the case of load decreasing. A load decrease of 50% occurs at 20 seconds of the simulation time as shown in Figure 6.11 (a), the control produces a corresponding reference speed given in Figure 6.11 (b). The system experiences a speed increase and voltage increase due to the sudden load drop as shown in Figure 6.11 (c) and (d). The controller then acts to change the input power of the system and bring the system to the reference speed, returning the voltage to the normal level within 20 seconds.

Figure 6.12 shows the load increasing case. A load increase from 50% to 100% occurs at 20 seconds of the simulation time as shown in Figure 6.12 (a), the corresponding reference speed is shown in Figure 6.12 (b). The system experiences a speed and voltage decrease due to the sudden load increase shown in Figure 6.12 (c) and (d). The controller performs the control and brings the system to the reference speed and voltage to the normal level within 15 seconds.

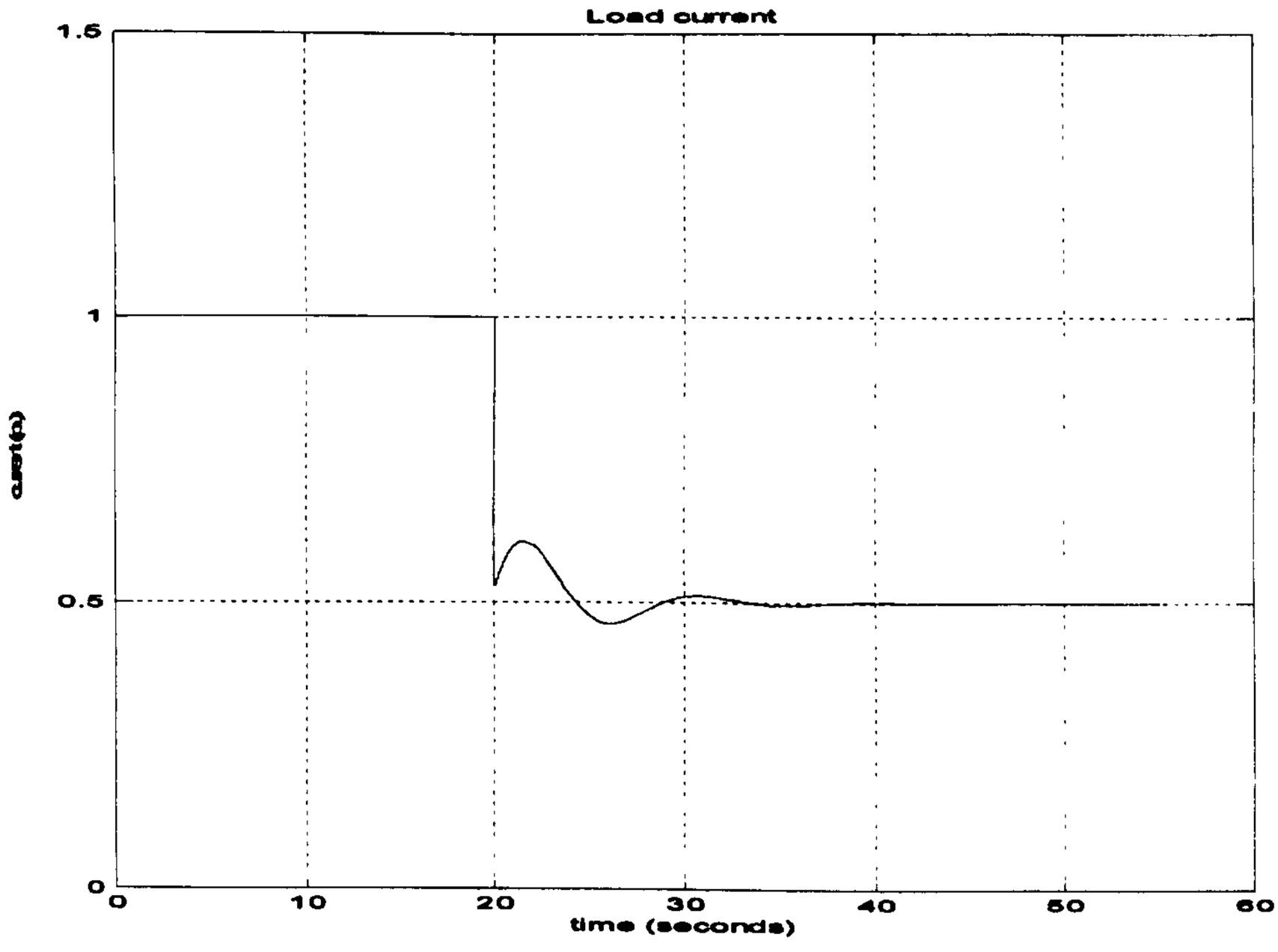


Figure 6.11. (a) Load current

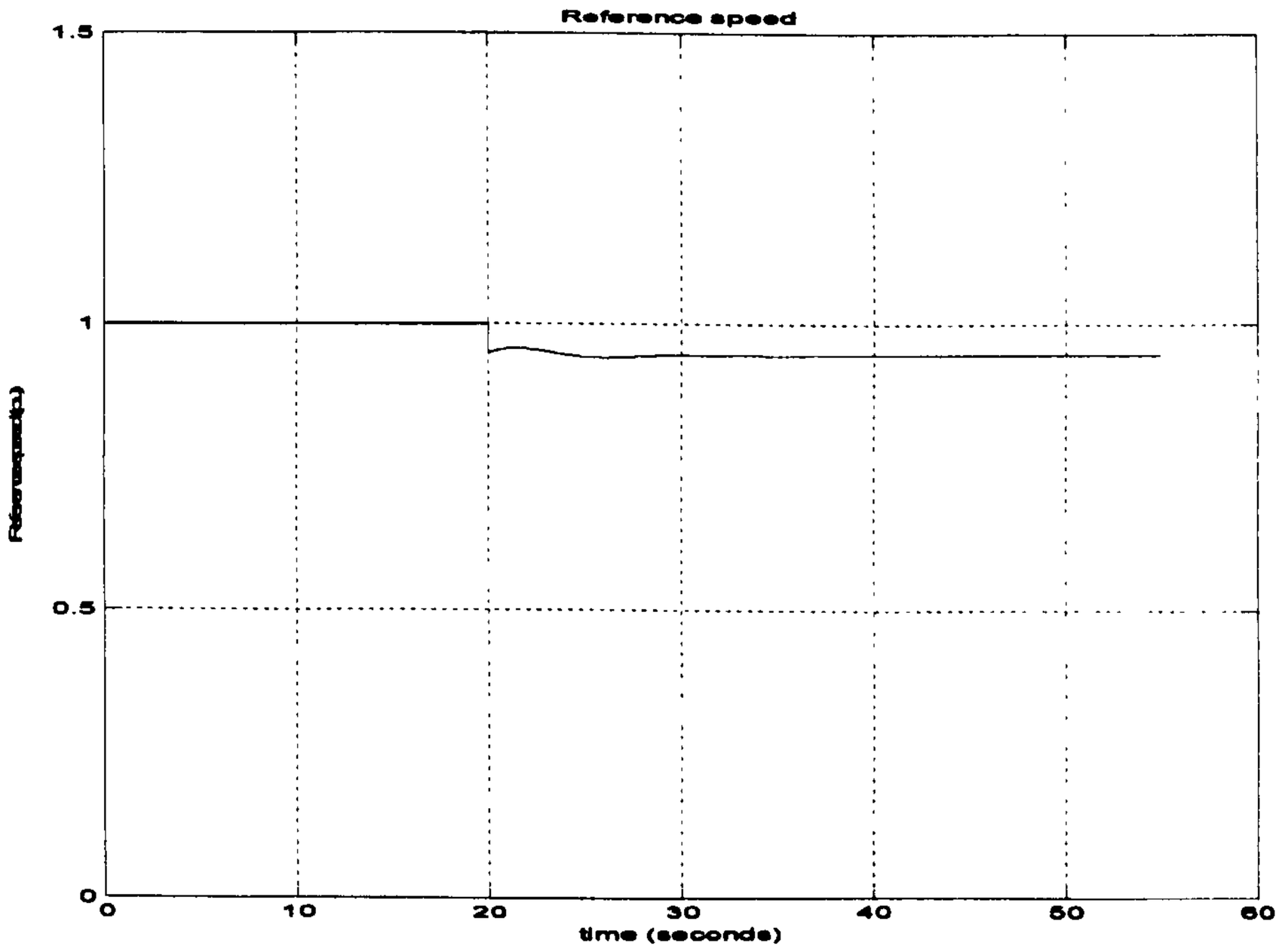


Figure 6.11. (b) Reference speed

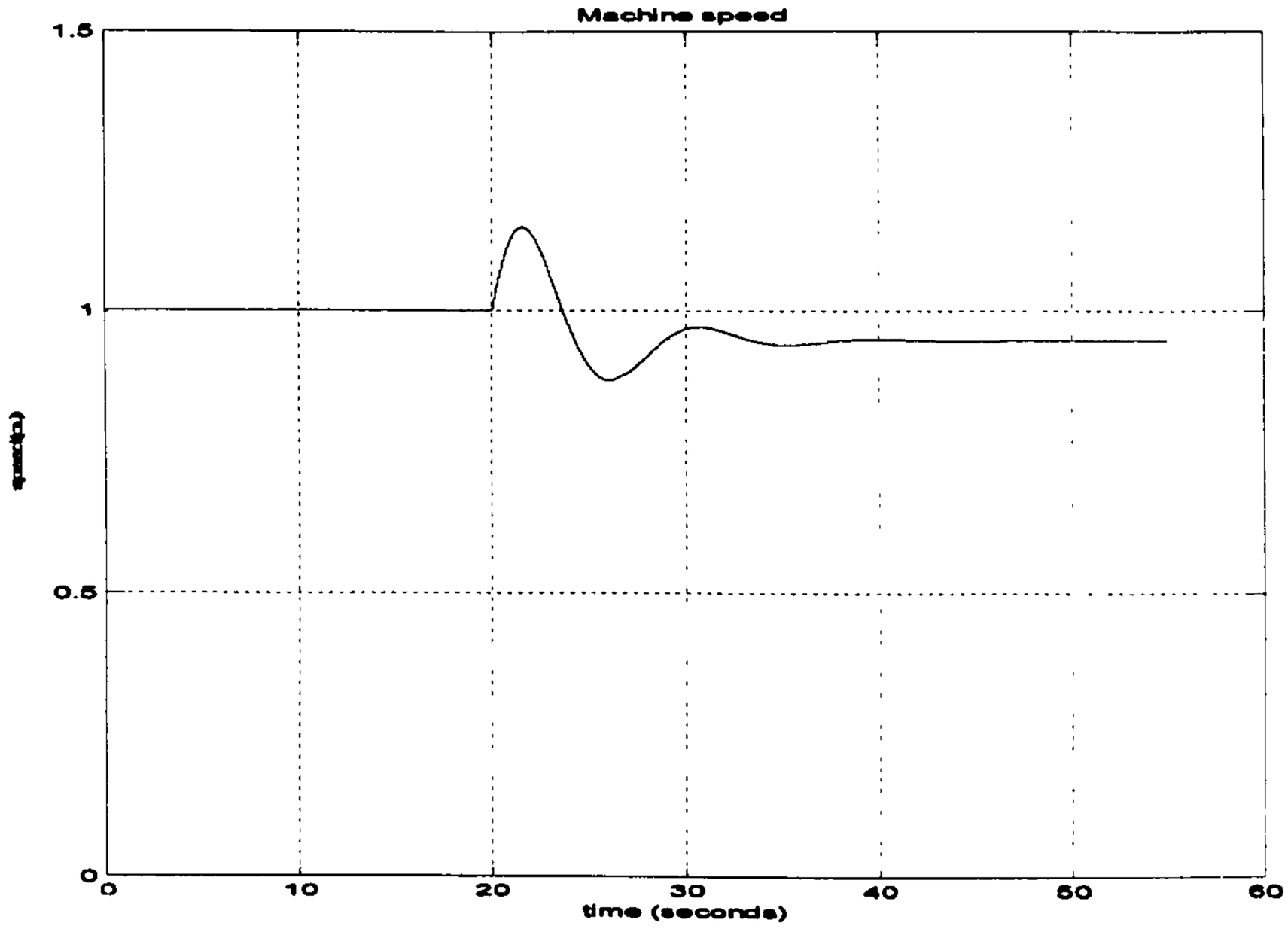


Figure 6.11. (c) Machine speed

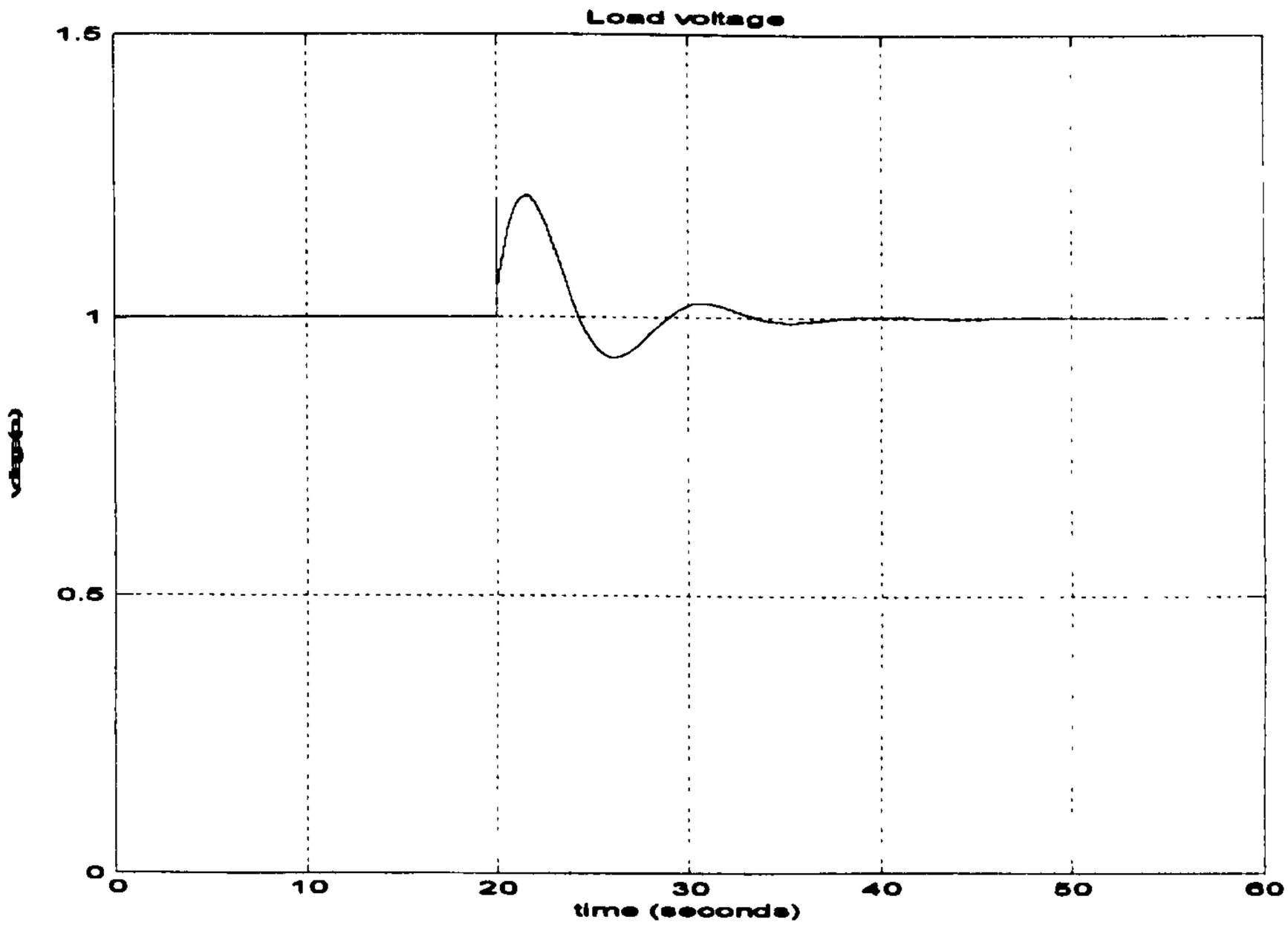


Figure 6.11. (d) Load voltage

Figure 6.11. Simulation case study (load decrease)

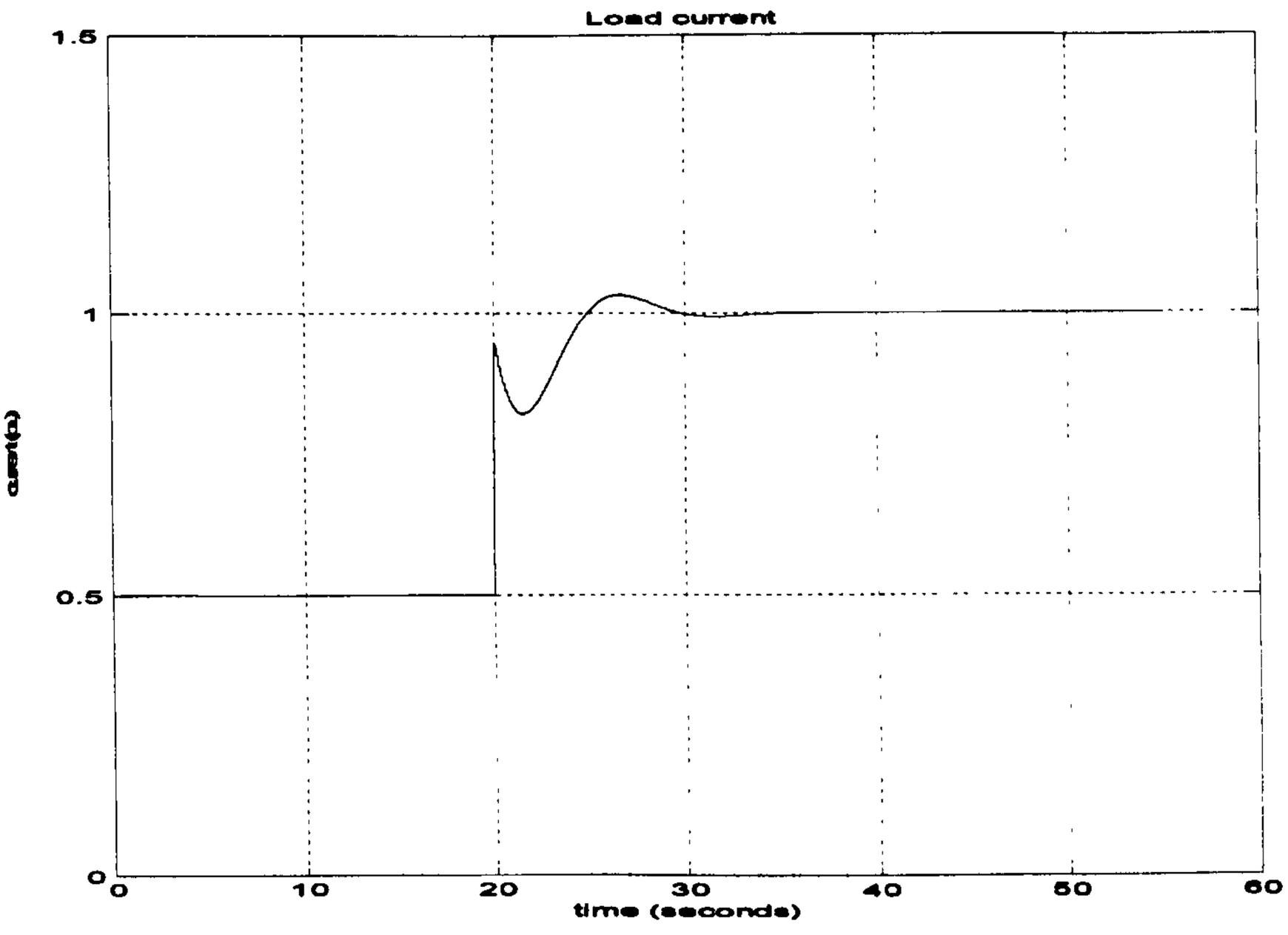


Figure 6.12. (a) Load current

Figure 6.12. (b) Reference speed

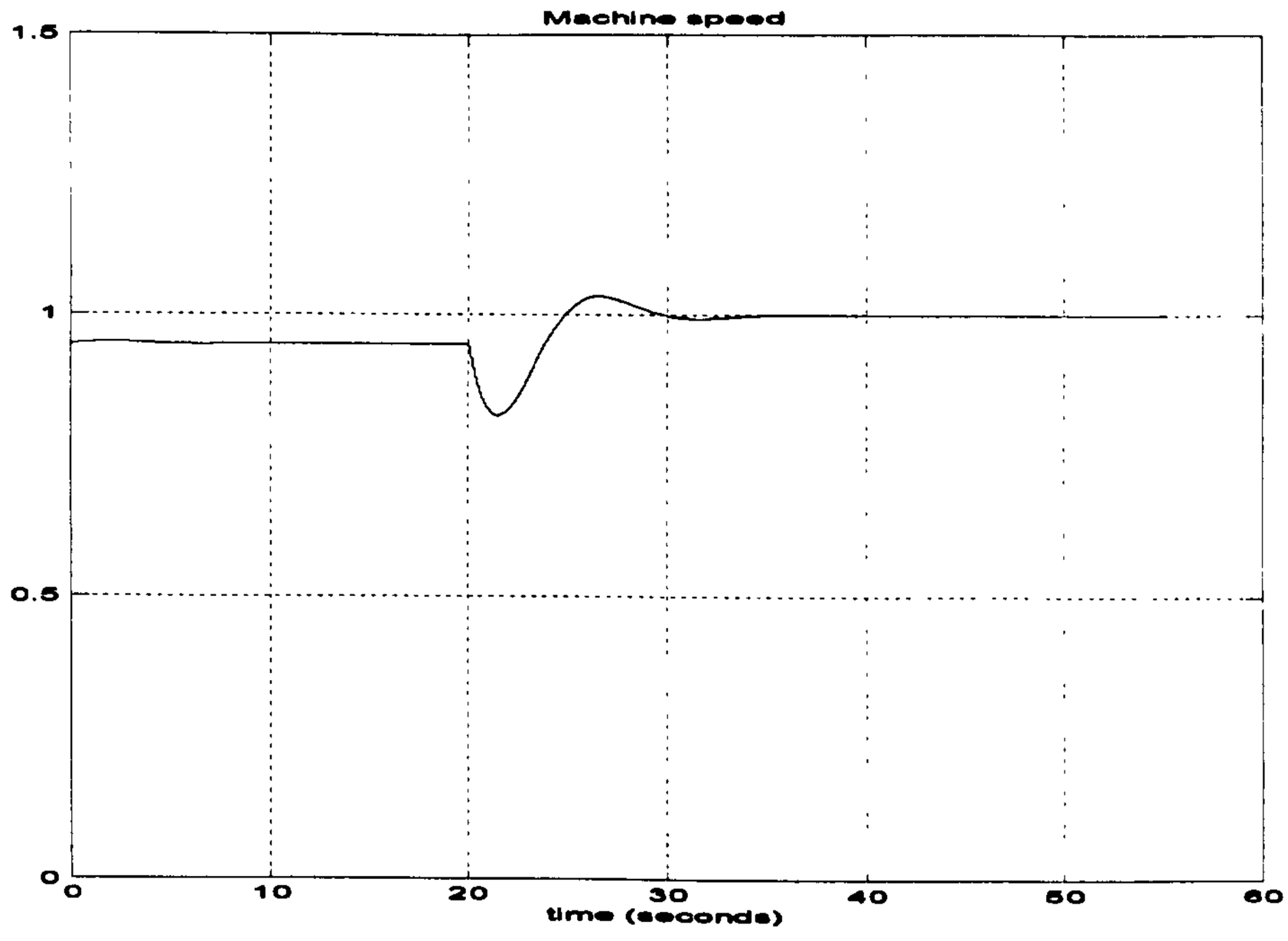


Figure 6.12. (c) Machine speed

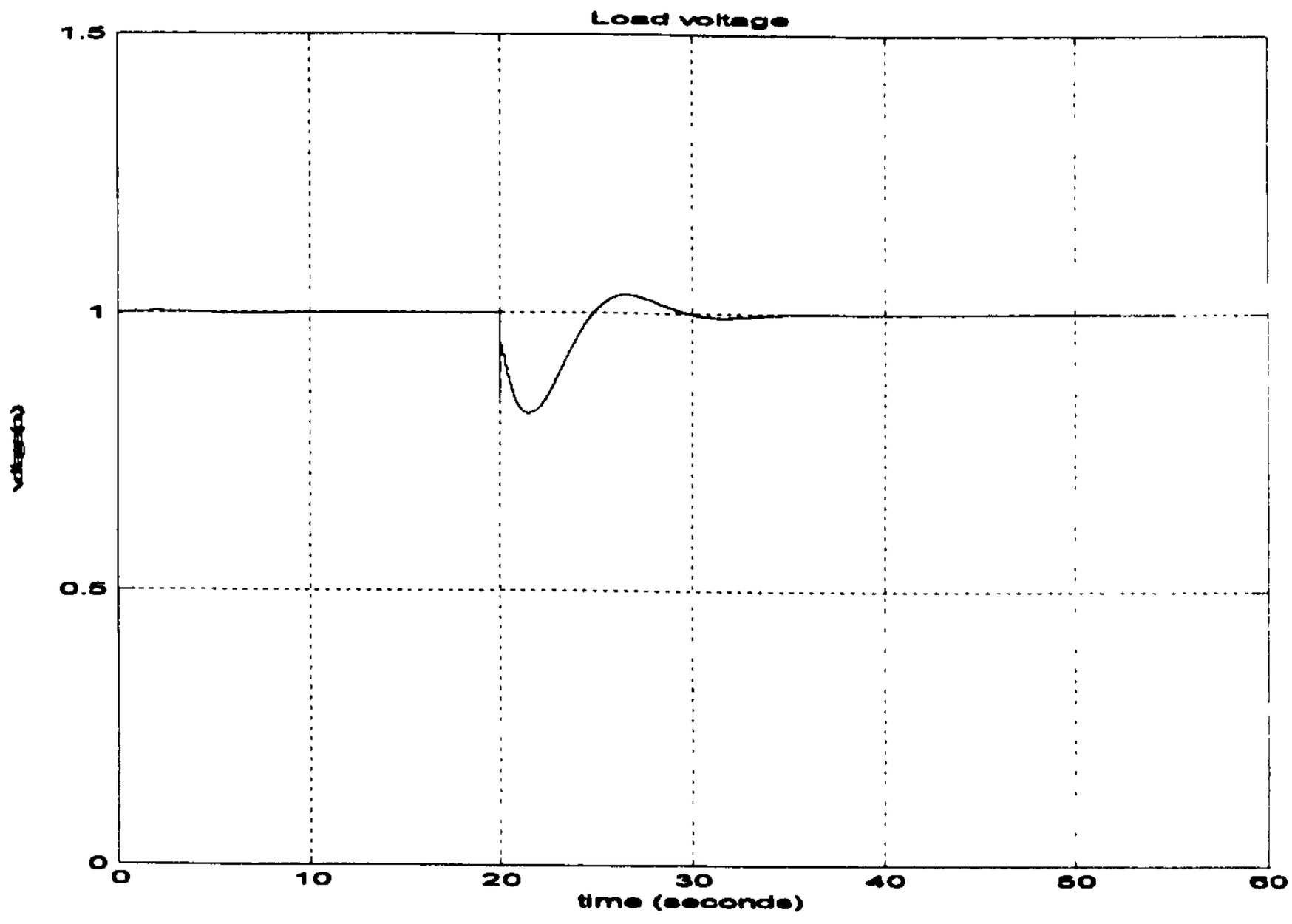


Figure 6.12. (d) Load voltage

Figure 6.12. Simulation case study (load increase)

Unlike the classical design approach, which requires exact mathematical models and precise numerical values, a basic feature of the fuzzy logic core is a process that can be controlled without detailed knowledge of the controlled systems. The control strategy can be expressed by a set of rules that describes the behaviour of the controller using linguistic terms. Proper control action can be inferred from this rule base that emulates the role of the human operator or a benchmark control action. Thus, fuzzy logic method is suitable for non-linear, dynamic processes for which an exact mathematical model may not be available.

The results have shown the effectiveness of the proposed control strategy and the hybrid variable speed controller under load changing condition. It is also noted that the transient of the system is much longer than the transient of the power electronic converter discussed in Chapter 5 due to the inertia of the mechanical system. If the required power changes, the system needs to adjust its fuel input accordingly. The variable speed controller responds to the load change bringing the system into a new stable operation point by making the adequate adjustment on the input power.

An important feature is that the variable speed controller adopts a fuzzy logic core. That has significantly reduced the complexity than using an arithmetic model, because it does not require a highly accurate modelling. The features of simplicity and less intensive mathematical calculation requirements enable the speed controller to be implemented using inexpensive hardware technology. A low-cost FPGA technology is used to implement the fuzzy logic based variable speed controller. The FPGA implementation of the controllers is discussed in a later chapter.

This chapter has presented the hybrid control method and the designed fuzzy logic core for the variable speed controller based on the principles and design methods for controls of fuel and power electronics in chapter 3 and 5. The next chapter presents a combined control strategy integrating power electronics system control and generator speed control.

Chapter 7

The System Control

This chapter mainly focuses on the overall system control. Firstly, the operational problems of small scale stand alone generation systems are discussed, then the control strategies for the generation system are proposed. Finally, the modelling and performance studies of the proposed control system are presented.

7.1. The operation of PM generator based stand alone generation systems

In a stand-alone generation system, keeping the output voltage at a constant predetermined level while the load changes is a strict requirement as a stand-alone generating system differs from a network connected system. The latter can be considered as an infinite power source, consequently, adding or shedding load does not significantly affect the system voltage and frequency. However, in a stand-alone generation system, any change of load or system configuration may represent a relatively big change that can significantly alter system operating conditions. The effects of large and sudden changes in the load need to be carefully assessed for small-scale generation systems.

Conventionally, both Automatic Voltage Regulators (AVRs) and speed governors are used with synchronous generators in stand alone generation systems to achieve constant speed and voltage operation. An Automatic Voltage Regulator (AVR) is a control device which automatically adjusts the output voltage of the generator to maintain the voltage at a relatively constant value. This is achieved by comparing the output voltage with a reference voltage and, based on the difference, necessary adjustments in the field current are made to bring the output voltage to the required

value. The speed governor is a device to control the speed by adjusting the power input from the prime mover and has been previously described in detail.

The fundamental difference between synchronous generators and permanent magnet generators is that synchronous generators can achieve voltage regulation by controlling the field excitation current. However permanent magnet generators normally have no adjustable excitation systems and obviously different methods must be used to regulate the system voltage. Basically the voltage of a permanent magnet generator may be adjusted by controlling the speed of the prime mover (engine) to produce a variable EMF. One fundamental concept to be considered is that in order to keep a system stable, the system has to balance the power supplied and the load plus power losses. The key aspect in controlling a stand-alone generation system is to produce the power demanded by the load. Therefore, whenever a load change occurs the generated power needs a corresponding adjustment quickly to balance the power requirement and maintain the system stability.

Obviously, the system may be operated more economically if the engine is operated at the most efficient point in the normal working range. This leads to the concept of variable speed generator systems in which the voltage and frequency are maintained constant using a suitable electronic interface between the machine and power systems.

To develop a complete control system for a stand-alone generation system with a permanent magnet generator, the analyses of the regulation relationship in a governed stand-alone system is considered. According to equation (3.5.a), the speed variation is determined by

$$\delta\omega_0 = {(C\delta q - \Delta G_L)}/{(Js + (A - B))}$$

The equation shows the system may be controlled to operate at almost constant speed ($\delta\omega_0 \cong 0$), or variable speed ($\delta\omega_0 \neq 0$) by adjusting the fuelling rate δq. For a stand-alone generation system carrying a significantly changing load in the normal operating range, fixed speed operation may not be efficient even though the load voltage can be

controlled at a constant level by a dc-dc converter. Alternatively, voltage regulation can be achieved by varying the engine speed, however the response is relatively slow as shown in Chapter 6. This chapter presents a combined control strategy to improve the system performance.

7.2. Control strategy for diesel engine generator set

In previous sections, the different operating requirements of field winding exciting and permanent magnet generators in stand-alone generation systems have been discussed. A method, using a chopper, to keep a constant dc link voltage for the inverter was discussed in Chapter 5. In addition, the fuzzy logic control method, which enables control of the operating speed to be included, was discussed in Chapter 6. The aim of the combined control system is simultaneously to meet the power required and at the same time, keep almost constant voltage and frequency.

In a small scale stand-alone system large and sudden load changes may represent changes that are a very high percentage of normal operating power. This may result in voltage transients of large magnitudes and create serious problems. For example, if the mechanical system alone is adjusted, the fuel input may not be sufficiently rapid because the responses of mechanical systems are slow. Nevertheless, the system may have to operate at a different power level after the transient period, therefore proper fuel input adjustment is necessary to balance the power.

In Chapter 5, the PWM scheme controlling the step-down chopper has been discussed. This method can be co-ordinated with the fuel governing mechanics to minimise the transient disturbance. The fuel control aims to maintain the stable and efficient operation of the system, while the dc link chopper control seeks to provide the voltage source inverter with an almost constant dc voltage input, so that a fixed modulation ratio can be applied to the VSI. The generator itself is de-coupled in frequency from the ac load, which receives the required power with relatively constant voltage amplitude and frequency. The control subsystems work together within an overall control strategy for the stand-alone generation system.

Figure 7.1 shows the proposed overall control strategy for the stand alone generation system. The complete control system comprises the variable speed controller and the control systems of power electronic converters. The variable speed controlled system provides an appropriate driving power to meet the demands of the changing system load, while the dc-dc conversion system absorbs voltage transients and maintains the voltage at the dc terminal of the inverter within the desired range, the frequency of the ac loading system being maintained by the inverter.

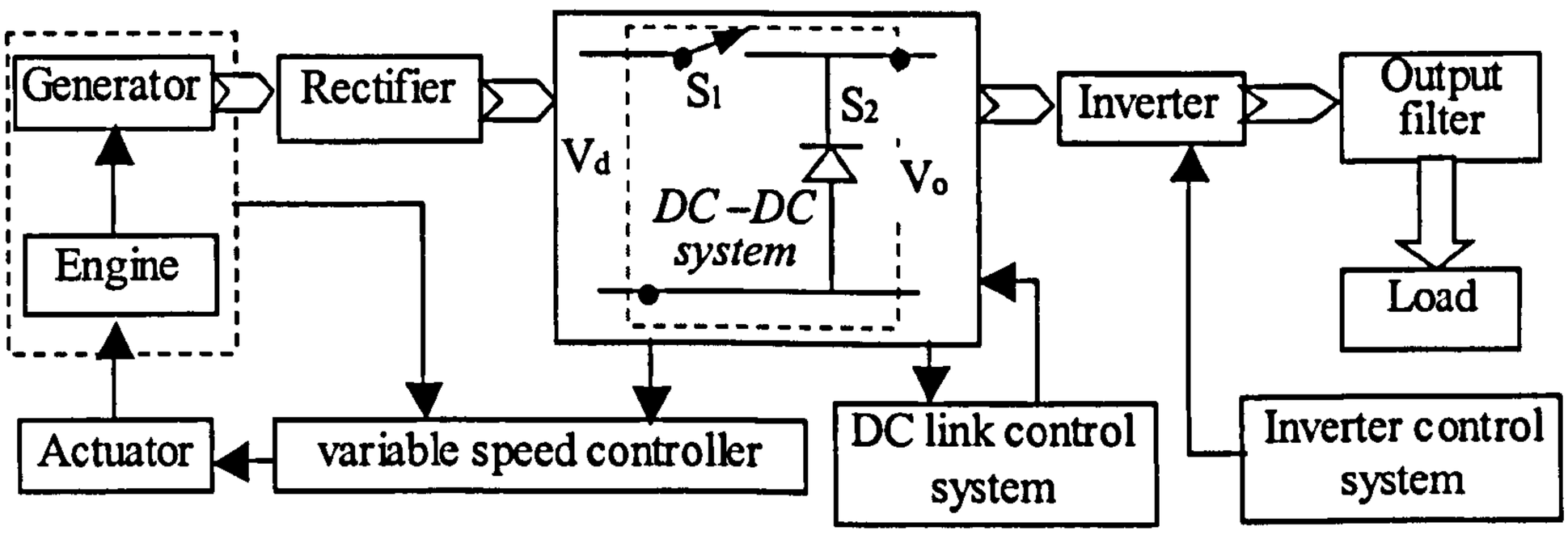


Figure 7.1. Control strategy for a stand-alone diesel engine generator system

7.2.1. Variable speed control principle

For simplicity, a simple circuit used to illustrate the compensation of voltage drop in the circuit is shown in Figure 7.2.

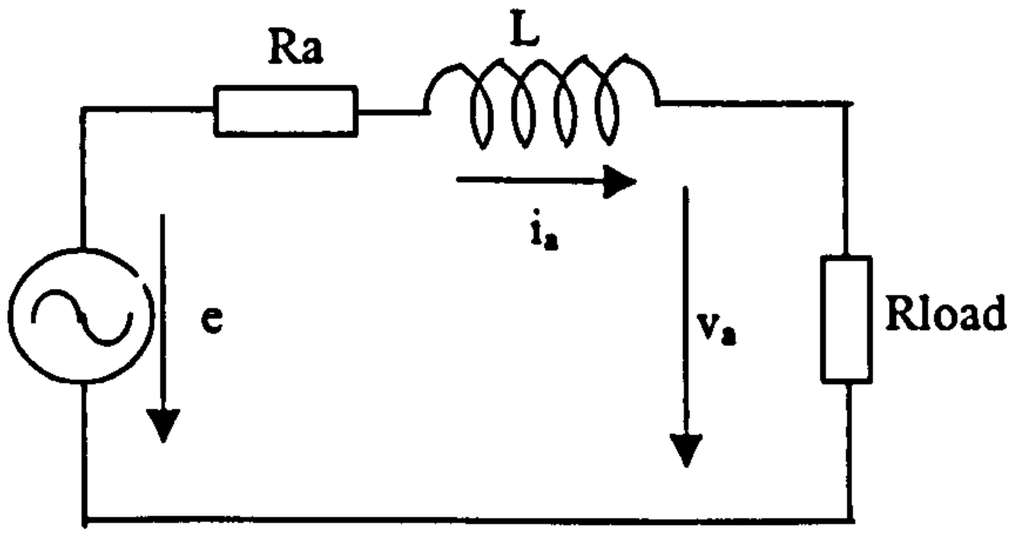


Figure 7.2. Circuit representation

The voltage relationship can be described by the equation:

$$e = v_a + L\frac{di_a}{dt} + R_a \cdot i_a \tag{7.1}$$

where

e is the electromotive force generated in generator stator winding, which depends on machine design and operation conditions such as the air gap flux density and the rotor rotating speed,

v_a is the generator terminal voltage, $v_a = R_{load} \cdot i_a$,

R_{load} is the equivalent load at the generator terminal. Here a pure resistive load is used.

L is the equivalent inductance of generator stator winding,

i_a is the current of the generator,

R_a is the equivalent resistance of the generator stator winding.

Assuming the generation system provides a constant voltage (Δv_a =0) to the load, then when a system load change ΔR occurs, a current Δi_a, which is equal to $V_{constant}/\Delta R$, will be produced and a corresponding power variation, $\Delta P = \Delta i_a V_{constant}$, occurs. Using the incremental form, equation (7.1) can be expressed as:

$$\Delta e = L\frac{d(\Delta i_a)}{dt} + R_a \cdot \Delta i_a \qquad (7.2)$$

Expressing in Laplace form equation (7.2) becomes:

$$\Delta e(s) = (Ls + R_a) \cdot \Delta i_a(s). \qquad (7.3)$$

That is, Δe is proportional to the current change. Equation (7.3) shows that as load current changes, a change of generated EMF, Δe, is able to keep a constant voltage in the simplified circuit. In an electrical machine system, Δe is proportional to the air gap flux and the rotational speed. If K_ϕ is used to represent the air gap flux factor, then:

$$\Delta e(s) = K_\phi \cdot \Delta\omega(s) \qquad (7.4)$$

Substituting (7.4) into (7.3) gives

$$K_\phi \cdot \Delta\omega(s) = (Ls + R_a) \cdot \Delta i_a(s)$$

and rearranging,

$$\Delta\omega(s) = \frac{(Ls + R_a)}{K_\phi} \cdot \Delta i_a(s) \qquad (7.5)$$

Assuming a constant output voltage then the change of Δi_a reflects the change of load power. Equation (7.5) indicates if a speed change, $\Delta\omega$, is generated in response to a

load change of Δi_a, then the system will tend to operate at a new stable condition (new working point) with a constant terminal voltage and balanced power.

If the system is operating at a speed ω_1 before the load changes and the new operating speed is ω_2 after the load changes, then ω_2 is equal to $\omega_1 + \Delta\omega$ and $\Delta\omega$ can be found according to the Equation (7.5). Where a diesel engine acts as the prime mover, the different operating speed can be achieved by adjusting the fuel-input. From the point of view of power balance, the generation system provides the power needed by the system, however the time taken to achieve the condition is unacceptable as shown in Chapter 6.

7.2.2. Power electronic conversion system control

In Figure 7.1 the power electronic system in the stand-alone generation system provides an effective solution to the voltage control. In the proposed scheme, the ac voltage produced by the generator is rectified into dc and the dc-dc converter provides the controlled dc voltage to the inverter. The regulated dc voltage is then converted back into ac using a PWM (Pulse Width Modulation) inverter. This type of conversion can be found in wind energy conversion systems [7][48][52].

The power electronic systems required have been discussed in previous chapters, however, the converters and associated control methods are briefly summarised in this section to highlight their functions in the integrated system.

In the system, the rectifier is an uncontrolled rectifier, so that the voltage at its output terminal changes while generator output voltage varies. The use of an uncontrolled rectifier can reduce the cost and complexity of the control system.

The dc-dc converter functions as a dc transformer, its ratio varies with its input voltage. A variable ratio dc-dc conversion stage allows the diesel engine and generator to be operated at variable speed, i.e. allows a changing voltage at dc link rectifier

terminal, but the voltage on the inverter side remains almost constant. The controllable dc-dc conversion stage decouples the frequencies of the two ac systems.

The dc to ac conversion in this project is achieved using a three-phase bridge power electronic inverter. The inverter is connected to the output terminal of the dc link to provide the desired ac voltage and frequency.

In summary, the rectifier in the system is a non-controlled device, the dc-dc converter has a feed back control system and the inverter is controlled by a fixed frequency and amplitude modulation switching strategy. With the combination of a dc-dc converter with variable conversion ratio and a voltage source inverter with the fixed modulation ratio, an effective appropriate power supply is provided.

7.3. Overall system modelling and simulation

In the previous sections, the control strategy for each part of the generation system has been discussed, the combined co-operative control is the key point of the control strategy for a stand-alone diesel engine driven permanent magnet generator system.

The complete system consists of three main parts:

- A closed-loop control system for the prime mover and generator system to perform the control of the speed and the input power.
- A closed-loop control system for the dc-dc voltage converter to regulate the dc link voltage.
- A voltage source inverter with the associated PWM control system to provide the ac voltage with constant amplitude and frequency to the load.

With the developed models of the engine, generator, power electronic converters and the control systems, the overall system simulation can be performed. However, due to the great difference between the time constants of the mechanical systems and power electronic systems, the memory capability of the available computing facility is not

adequate to support the simulation of such a complete system. Therefore, the studied system is sub-divided into two sub-systems as shown in Fig. 7.3.

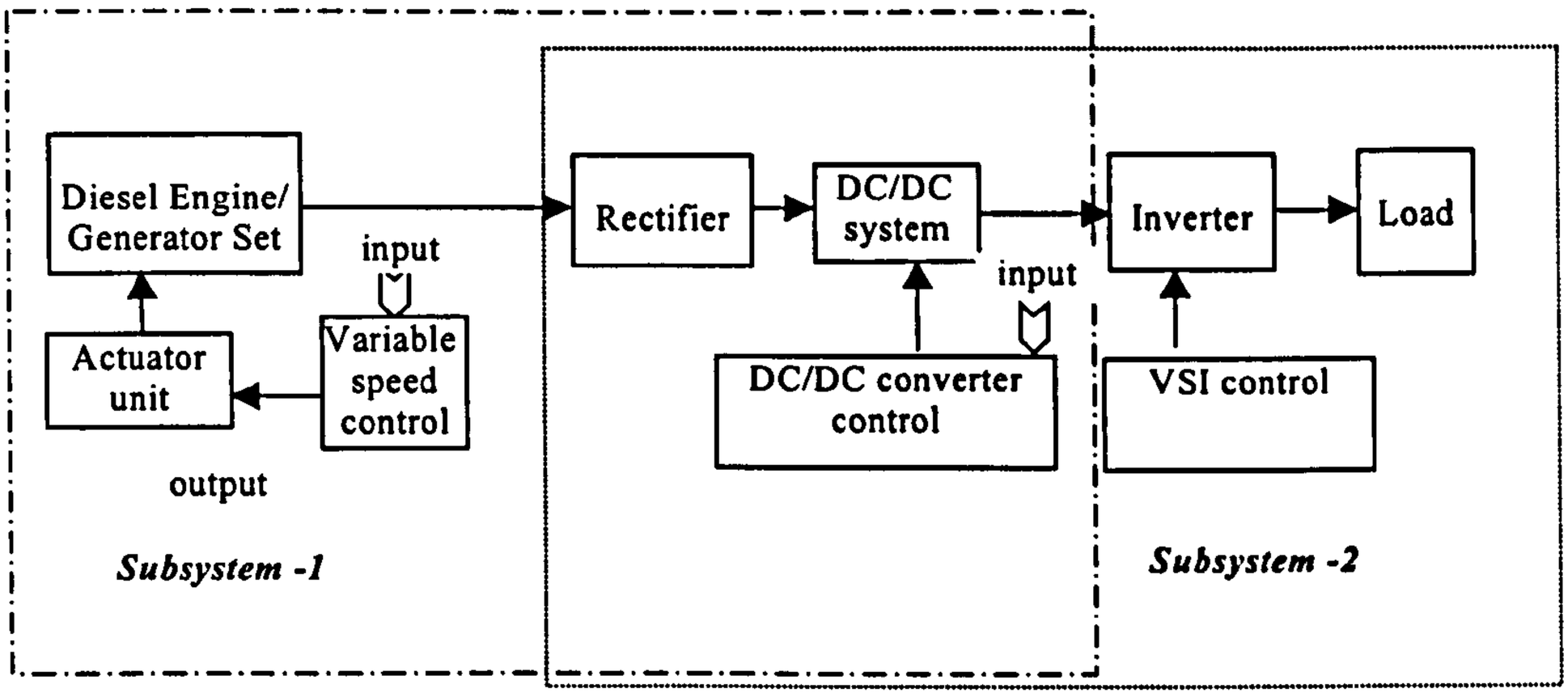


Figure 7.3. Block diagram of the simulation system

Subsystem-1: the diesel engine-generator system and part of power electronic system including ac-dc and dc-dc conversion, the speed controller and dc converter control. A resistive load is used to load the dc-dc converter.

Subsystem-2: the power electronic system composed of the rectifier, dc-dc converter, inverter and the associated control system for power electronics.

The overlap of the two subsystems is designed to enable the effectiveness of the control systems to be demonstrated.

The simulation results of the Subsystem-1 are presented in Figure 7.4. A load increase occurs at 1 second of the simulation time as shown in Figure 7.4 (a). Following the disturbance, the speed controller responds to set a new level of reference speed given in Figure 7.4 (b). The fuel input control is then adjusted. The action brings the system to the new reference speed as shown in Figure 7.4 (c), which also shows there is a short transient before the system reaches the new stable state after the load disturbance. It can be seen that the dc link voltage tends to change, however, the dc-dc converter keeps the chopper output voltage almost constant and the voltage transients are significantly reduced as shown in Figure 7.4 (d). As the speed varies, the generator

output voltage correspondingly changes. The magnitude and frequency of the ac input voltage to the rectifier system was increased with the increase of the machine speed is shown Figure 7.4 (e).

The simulation results of Subsystem-2 are presented in Figure 7.5. A load increase occurs at 0.14 seconds of the simulation time as shown in Figure 7.5 (a). A corresponding dc link current increase occurs as shown in Figure 7.5 (b). Figure 7.5 (c) shows that the dc-dc converter maintains the chopper output voltage almost constant. The ac voltage waveform of the PWM inverter in Figure 7.5 (d) shows that the ac loading system voltage magnitude and frequency are kept constant during the load disturbance. The magnitude and frequency of the ac input voltage to the rectifier system is assumed constant during the simulation as shown Figure 7.5 (e).

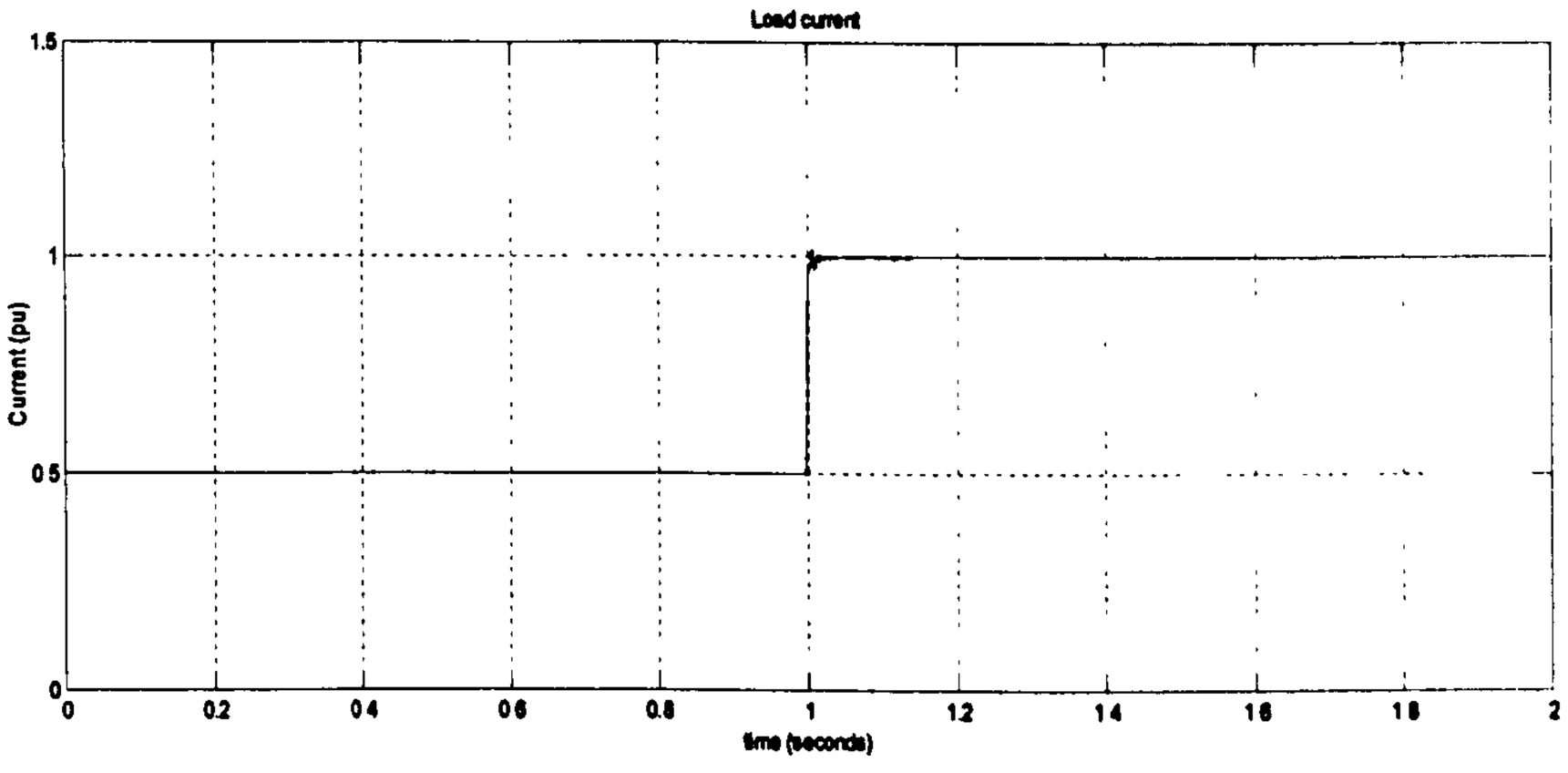


Figure 7.4. (a) Load current

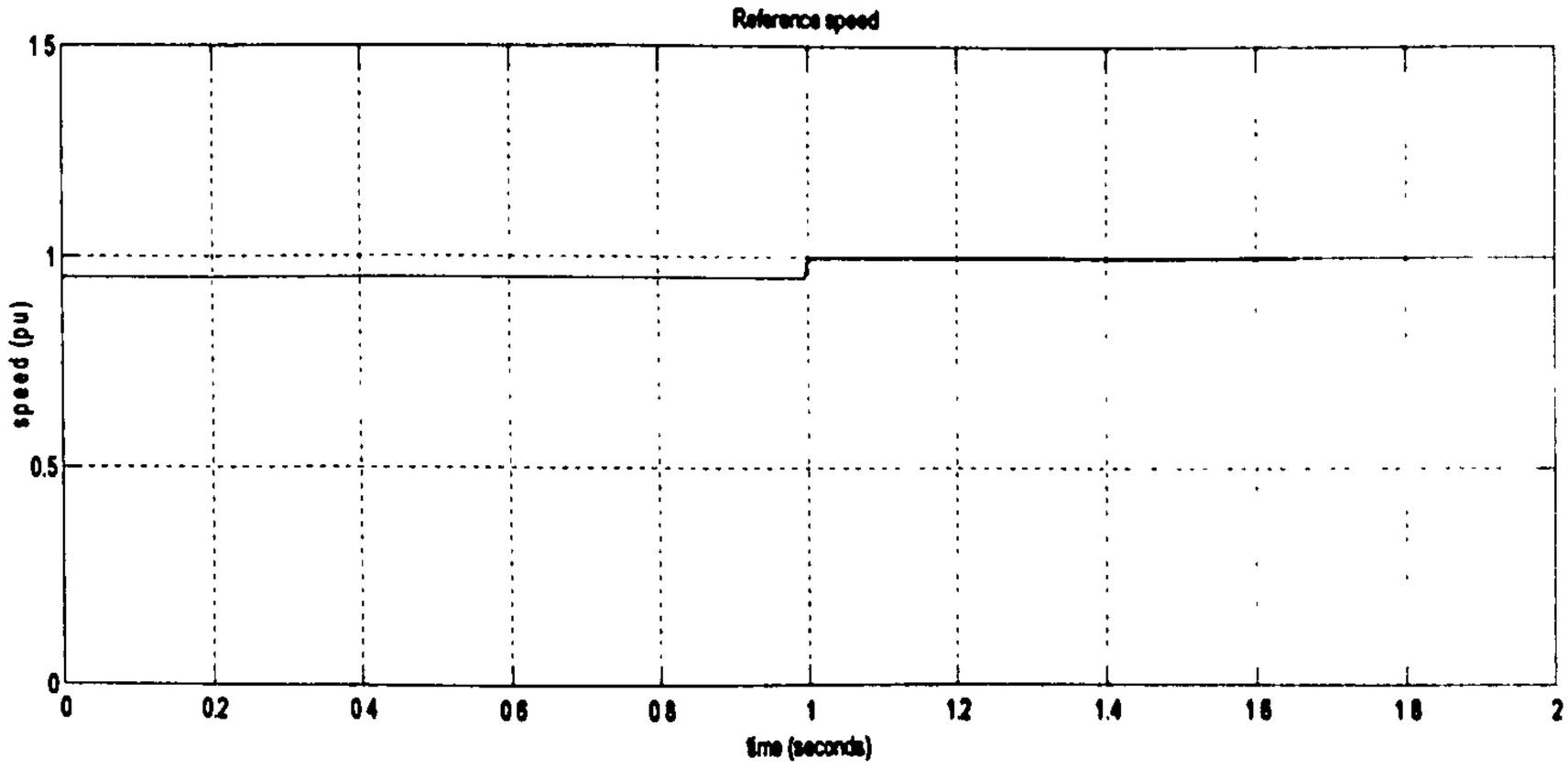


Figure 7.4. (b) Reference speed

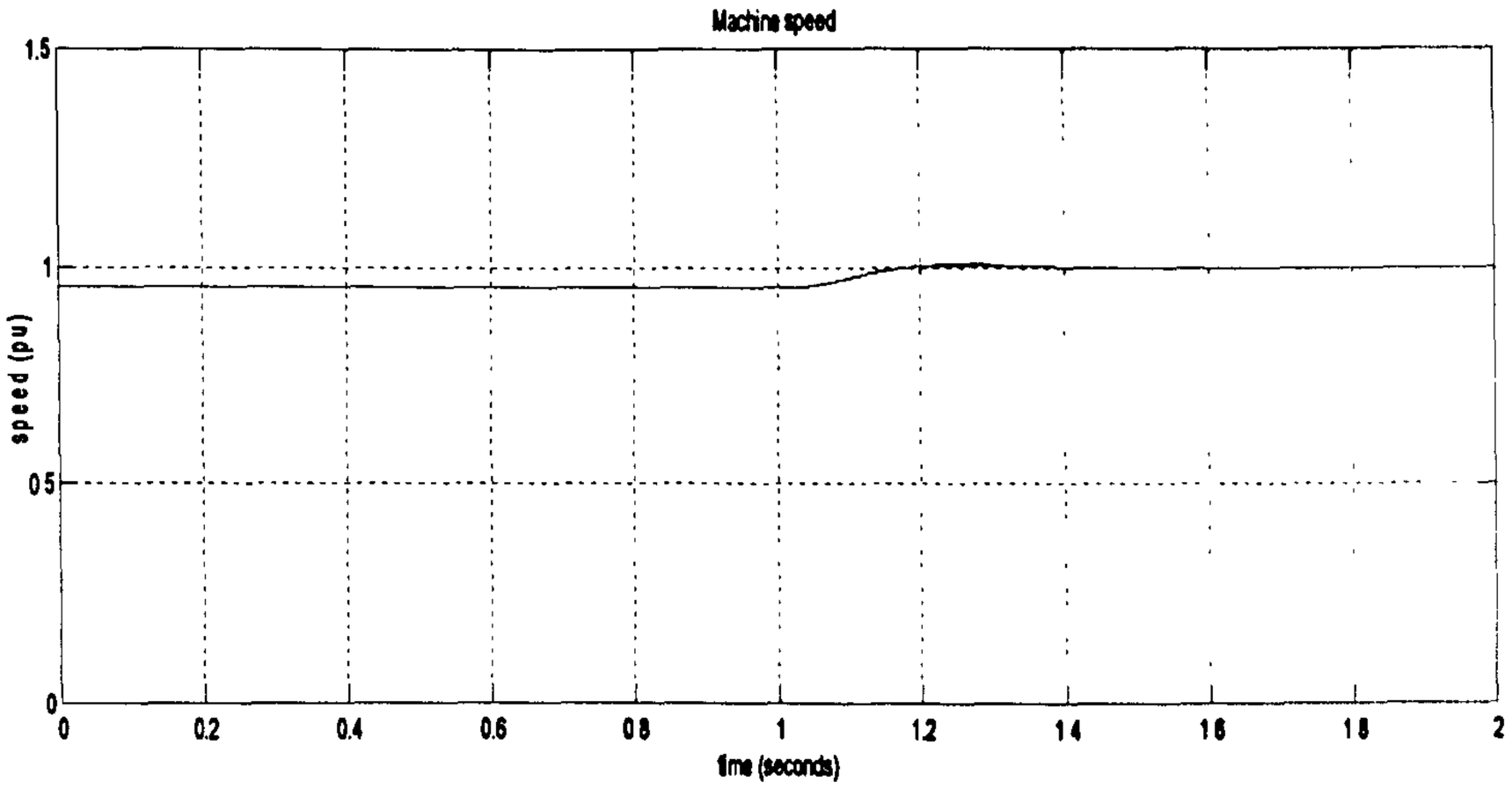


Figure 7.4. (c) Machine speed

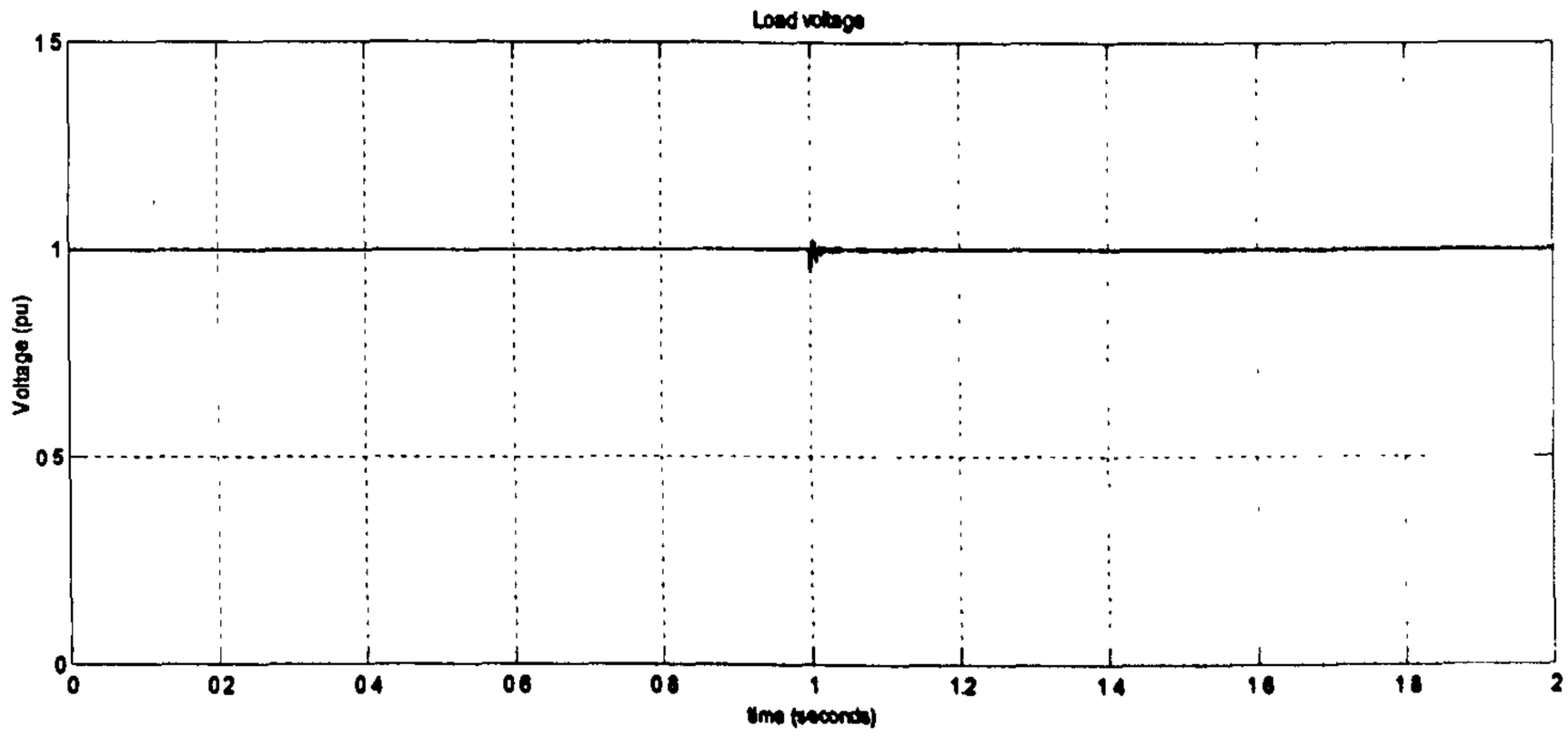


Figure 7.4. (d) Output voltage of dc-dc converter

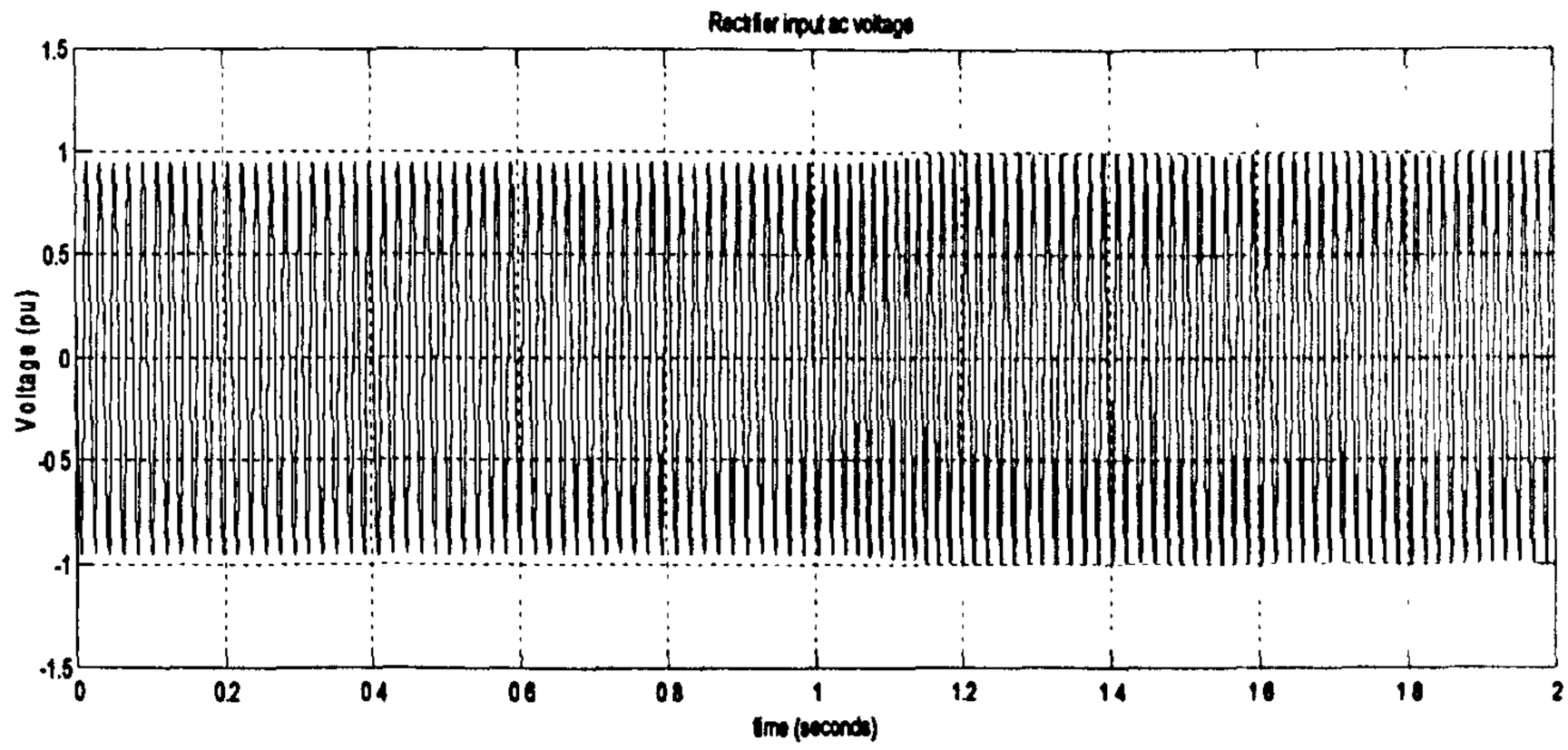


Figure 7.4 (e) Rectifier ac voltage

Figure 7.4. Simulation results of Subsystem-1

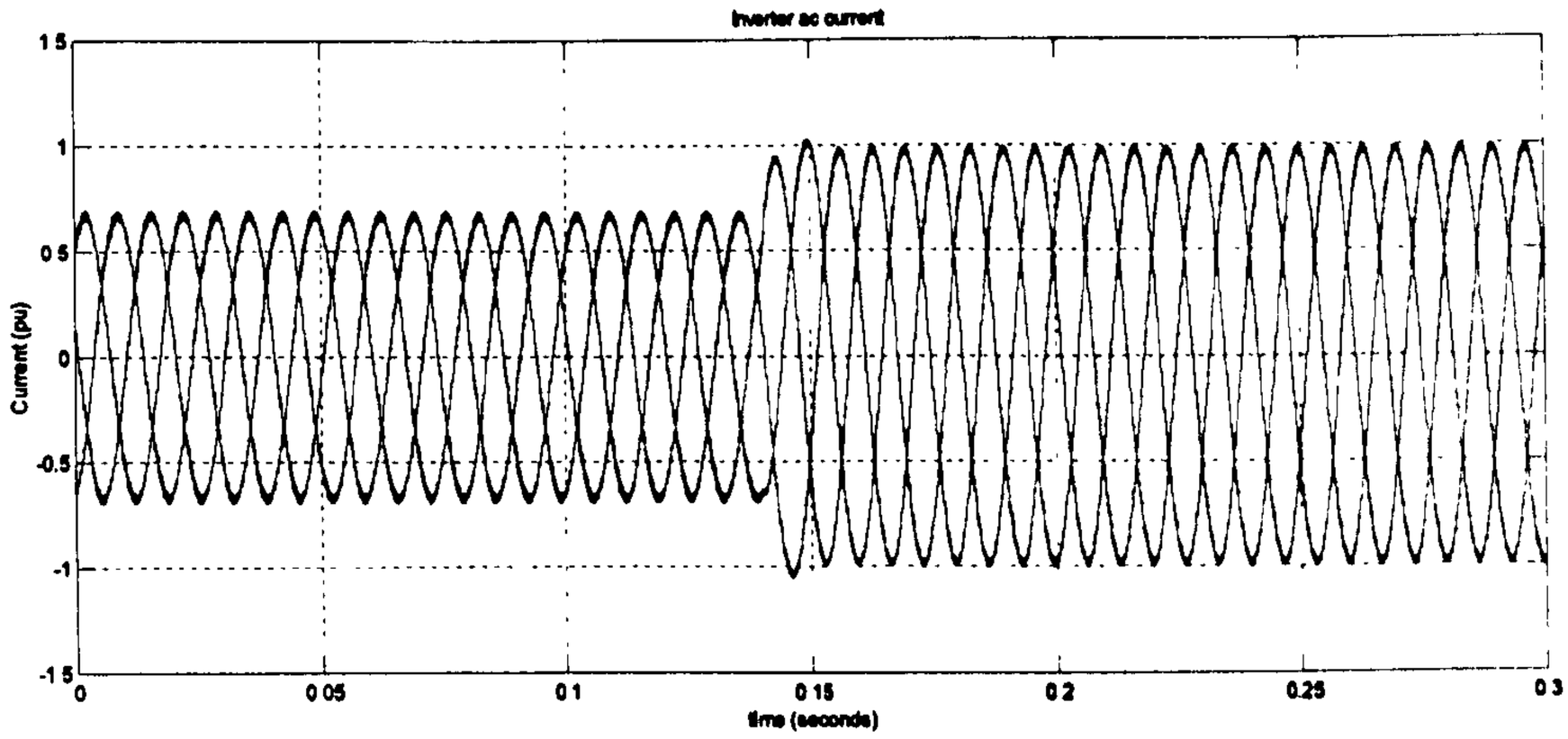


Figure 7.5. (a) Inverter output ac current

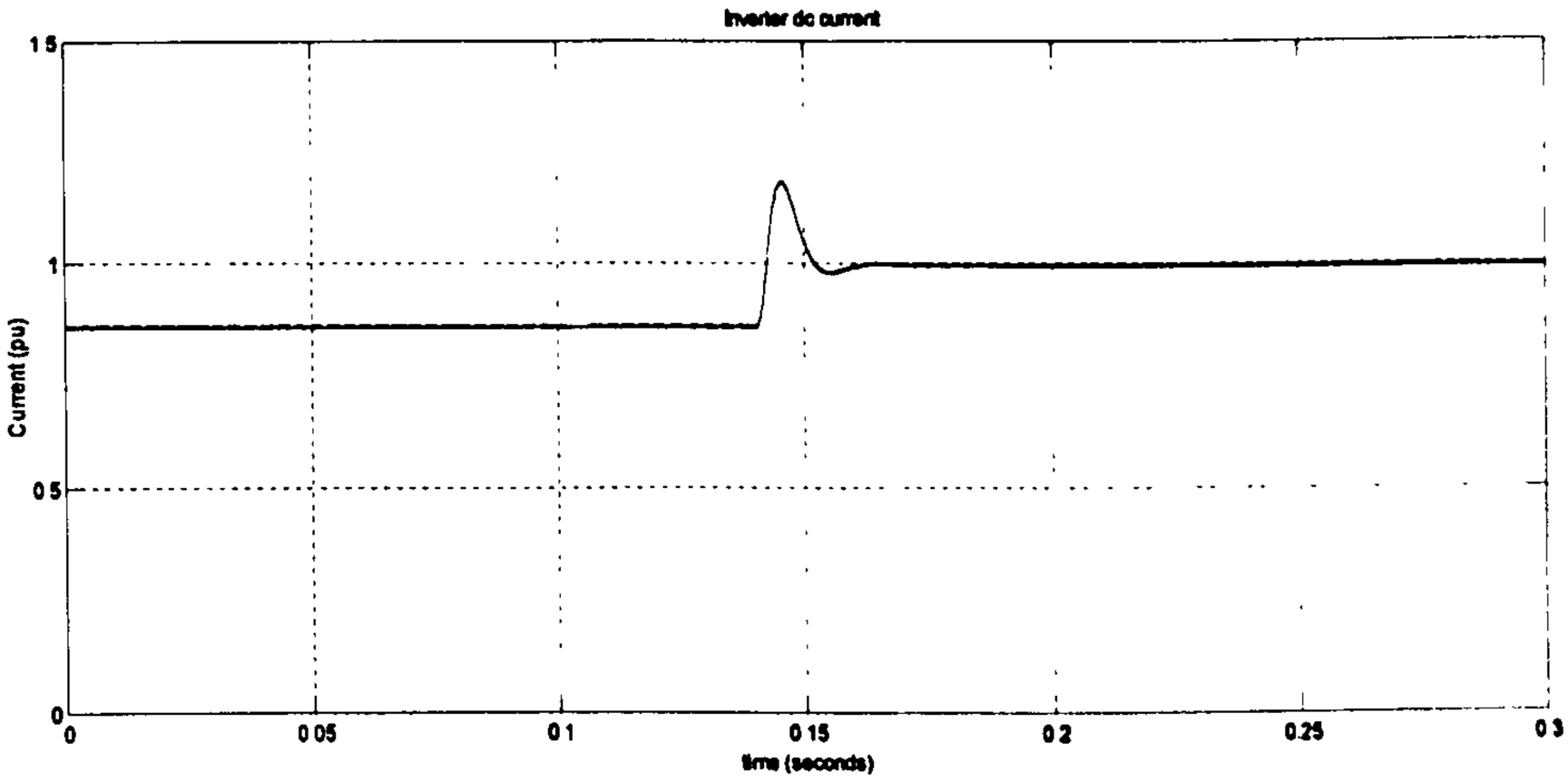


Figure 7.5. (b) Inverter side dc current

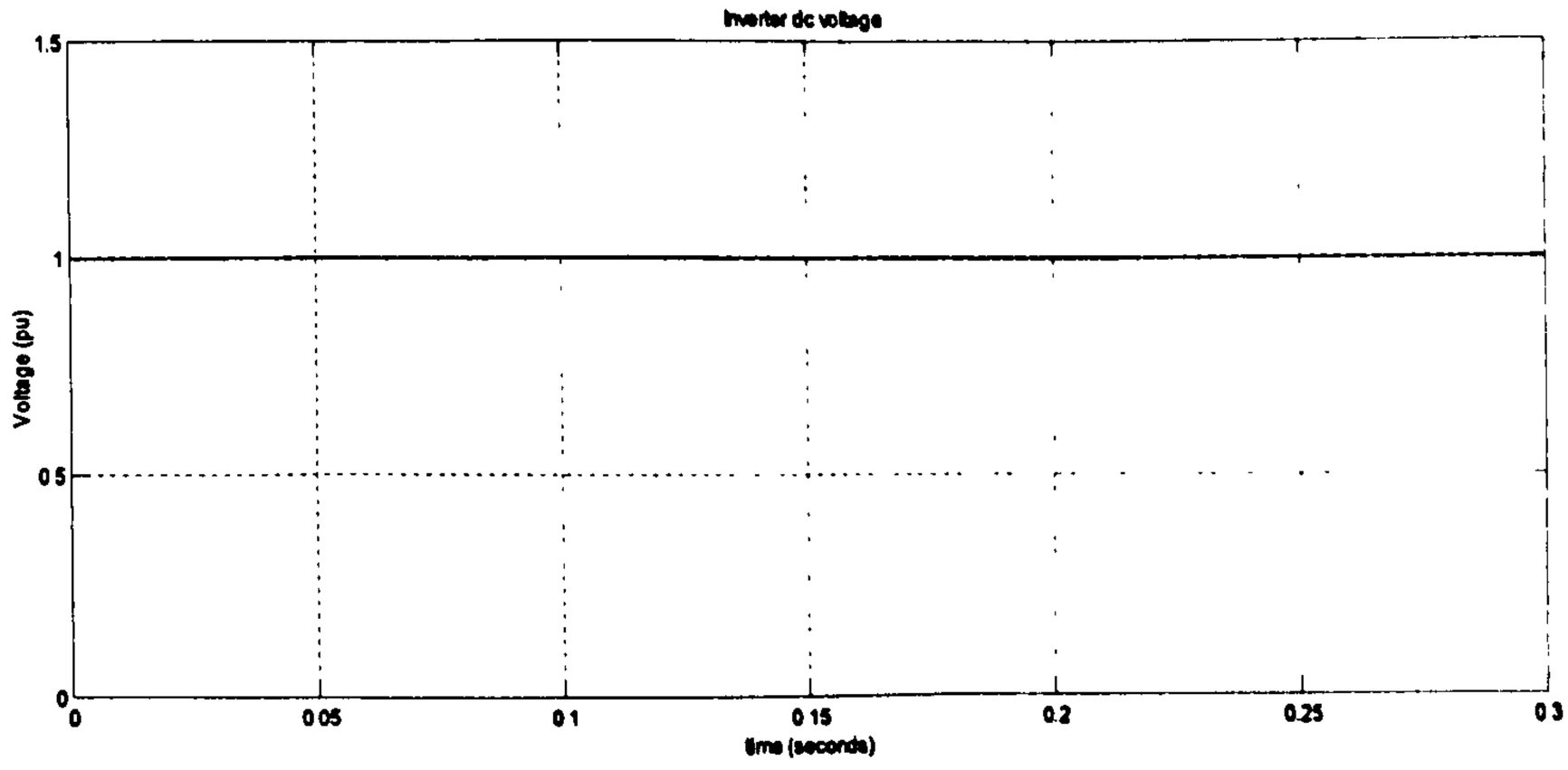


Figure 7.5 (c) Inverter side dc voltage

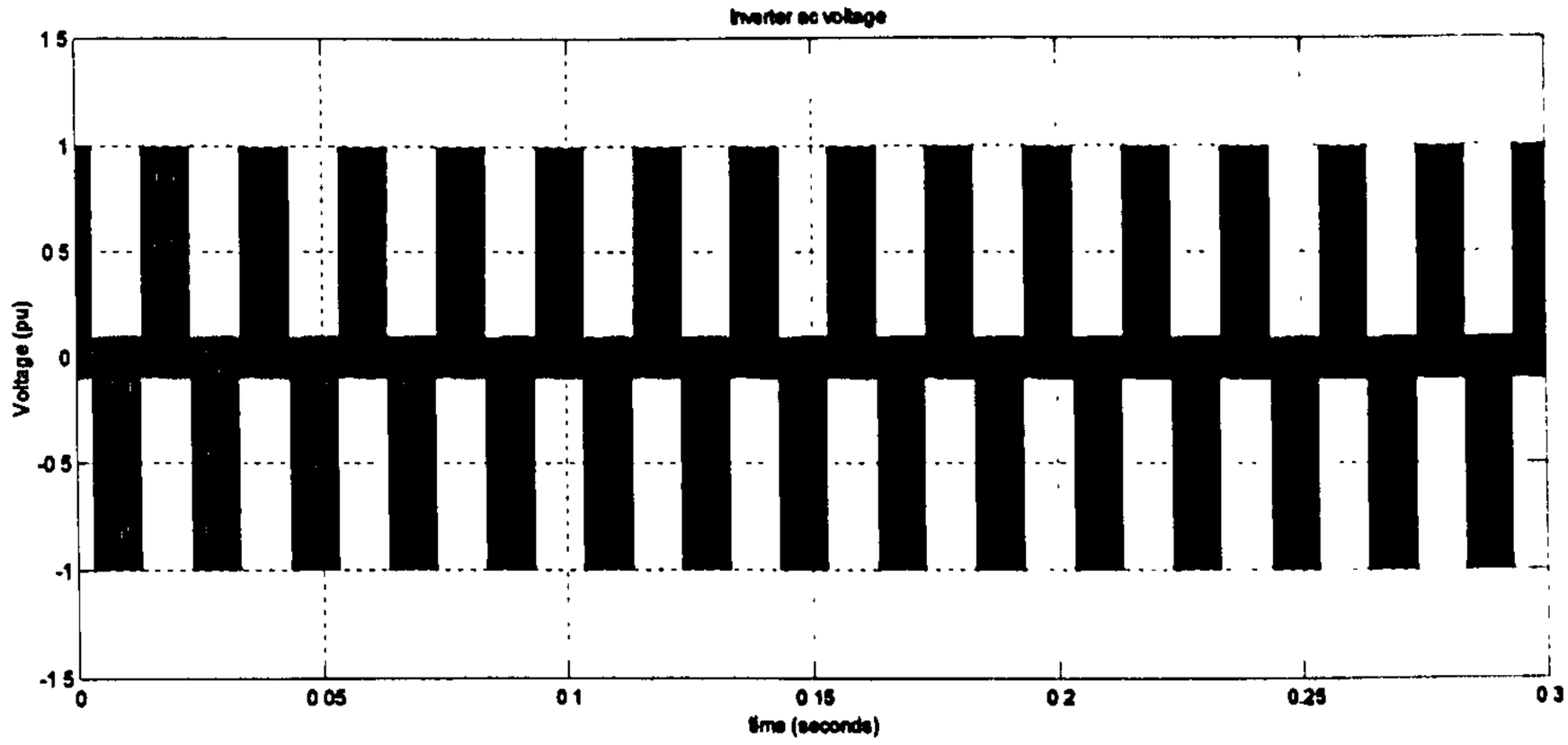


Figure 7.5. (d) Inverter output ac voltage

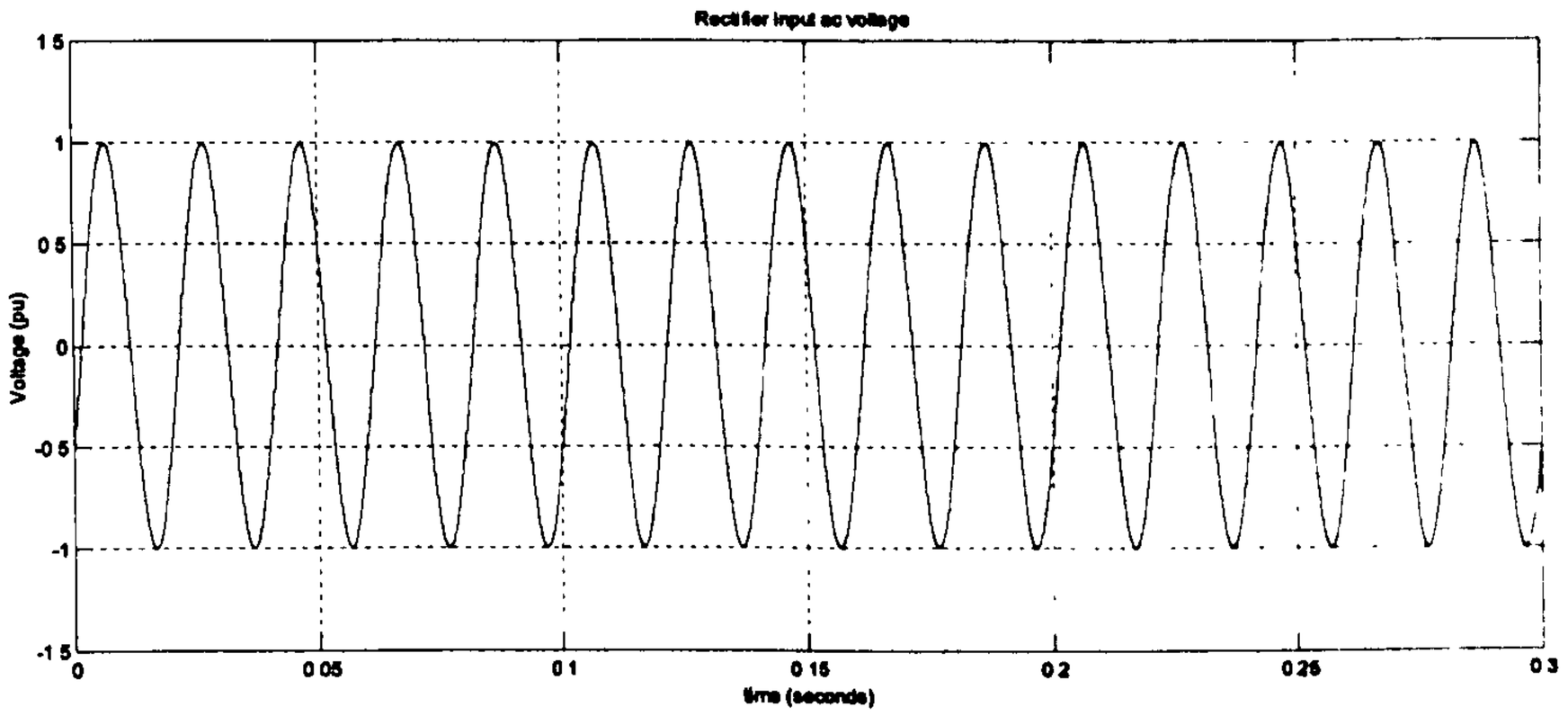


Figure 7.5. (e) Rectifier ac voltage

Figure 7.5. Simulation results of Subsystem-2

7.4. Discussions and summary

The control strategies have been developed by integrating two controllers (i) Fuel input control, (ii) Power electronic converter control. The investigation and evaluation of the system and control strategy have been performed by simulation. The simulation results show the proposed combined control strategy for the whole control system performs well.

In terms of the system power balance, as load increases, the speed controller adjusts the operating speed to a higher level so as to generate more power to balance the load

increase. In the proposed control strategy, keeping a constant dc link voltage is an important condition for providing an adequate voltage to the consumer in the ac system, this has been secured by a dc-dc converter and a fixed modulation ratio switching PWM inverter. The chopper controller can respond quickly to the system dynamics and therefore quickly provides control of the DC link voltage. The mismatch between the high speed system transient and moderate speed response of the speed control is compensated for by the chopper.

Simulation studies have been performed with separated subsystems and the expected results were obtained. It has been shown that the developed models and the controllers work effectively. The corresponding results show that the isolated generation system under a changing load can be controlled to operate at variable speed. If the required power changes, the system adjusts the fuel input accordingly, and eventually the desired operating speed. The variable speed controller responds to the load change and operates the system at a variable speed. The reference speed reflects the steady state fuel rate.

The results obtained have proven that the control strategy is effective for stand-alone generation systems. The control system implementation and experimental studies are presented in the following chapters.

Chapter 8

Control System CPLD/FPGA Implementation

Traditional control systems, often based on analogue and digital electronic circuits, usually have the advantages of fast dynamic response but have the drawback of complex circuitry, limited functions and difficulty in circuit modification. Many modern techniques are used in present control system applications, such as, EPROM supervisory controller, microprocessor based or DSP based digital controllers. These modern control schemes overcome most of the disadvantages of traditional electronic circuits and also have the important features, such as software control and flexibility in adaptation to various applications. The advancement of very large-scale integration technology and the development of application-specific integrated circuit technology have encouraged several new specialised technologies, in particular, Complex Programmable Logic Device (CPLD) and Field Programmable Gate Array (FPGA), which enable rapid prototyping of digital systems.

VHDL is closely related to FPGA and CPLD and is an industry standard modelling language for the design and description of electronic systems. Use of VHDL as a modelling and simulation tool in the design of control systems is a new development aimed at exploring and rapidly prototyping more compact, fast, and cost effective controllers. In this research the VHDL language is used as the tool for targeting the FPGA/CPLD chip implementation. This has led to the application of a "top-down" design methodology and logic synthesis tools. The designed controllers are to be implemented with programmable chips to minimise the circuit complexity. This chapter starts with a brief introduction of the CPLD and FPGA devices used; and the modelling of the CPLD and FPGA based hardware using the VHDL development tool: Xilinx Foundation Express. This chapter describes the CPLD/FPGA based control

system structures, covers the simulation studies and timing analysis of the VHDL models of the control systems and finally presents the results obtained.

8.1. Introduction of CPLD/FPGA technology and the VHDL development tool

8.1.1. Xilinx XC4000XL and XC9500 Series

The CPLD and FPGA devices (XC9500 and XC4000XL) used in this project are described in this section.

XC4000XL family

The Xilinx XC4000XL family, one of the Xilinx Field Programmable Gate Arrays series, has significant features for control system implementation, such as high-performance, high-capacity, unlimited reprogram ability. XC4000XL family FPGAs are implemented with regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), which are interconnected by a powerful hierarchy of versatile routing resources (routing channels) and surrounded by a perimeter of programmable Input/Output Blocks (IOBs). The layout is illustrated in Figure 8.1.

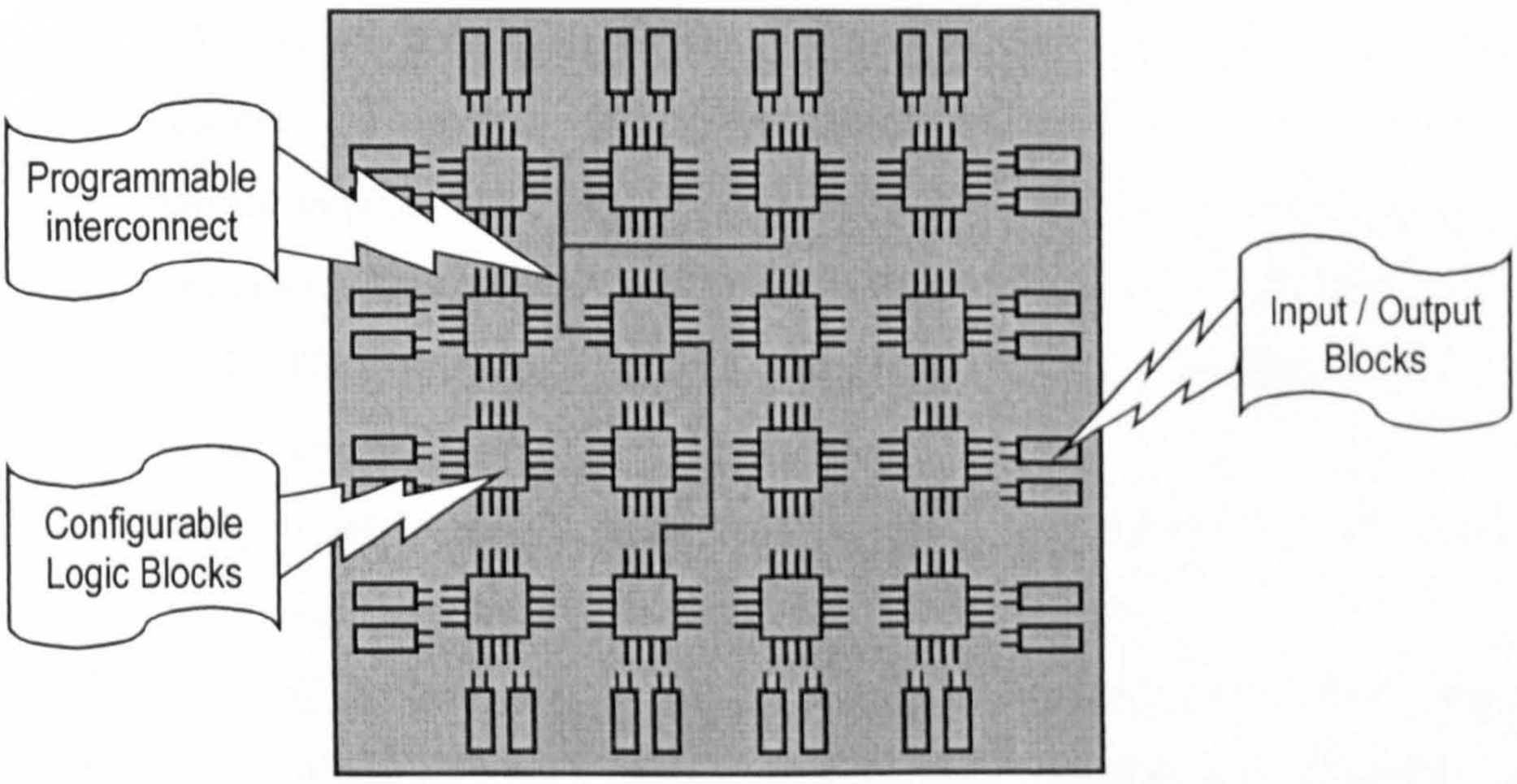


Figure 8.1. Basic FPGA block diagram

There are mainly three types of elements in a FPGA device [43].

- A matrix of Configurable Logic Blocks (CLBs). The CLBs provide the functional elements for implementing most of the desired logics. A net of CLB routing surrounds each CLB in the array and the routings are connected to the CLB inputs and outputs. Each CLB contains storage elements that can be configured as either flip-flops or latches. This capability makes the FPGA highly synthesis-compatible.
- Peripheral Input/Output Blocks (IOBs). User-configurable input/output blocks provide the interface between external package pins and the internal logic. Each IOB controls one package pin and can be configured for input, output, or bi-directional signals. Inputs on the XC4000XL are TTL and 3.3V CMOS compatible. The I/Os on the XC4000XL are fully 5-volt tolerant even though the supply voltage, Vcc, is 3.3V. This allows 5V signals to directly connect to the XC4000XL inputs. In addition, the 3.3V Vcc can be applied before or after 5 V signals applied to the I/Os. This makes the XC4000XL immune to power supply sequencing problems. Outputs on the XC4000XL are pulled to the positive supply rail.
- A net of metal interconnects. All internal connections are composed of metal segments with programmable switching points and switching matrices to implement the desired routing. As described at the beginning of this section, the XC4000XL FPGAs can be reconfigured an unlimited number of times. The interconnection can be programmed and reprogrammed such that the routing blocks can be optimised or re-composed. There are several types of interconnects in these FPGAs:
  1. CLB routing is associated with each row and column of the CLB array.
  2. IOB routing forms a ring around the outside of the CLB array. It connects the I/O with the internal logic blocks.
  3. Global routing consists of dedicated networks primarily designed to distribute clock signal throughout the device with minimum delay and skew.

The devices are customized by loading configuration data into internal static memory cells. The values stored in these memory cells and implemented interconnections determine the logic functions of the FPGA. XC4000 family devices support system clock rates around 80 MHz. The XC4000XL series product includes various devices, such as XC4005XL, XC4010XL, XC4013XL, XC4020XL, XC4028XL, XC4036XL, XC4044XL, XC4052XL, XC4062XL, XC4085XL, the maximum logic gate range is from 5000 to 85,000 [43].

XC9500 family

The XC9500 Complex Programmable Logic Device (CPLD) family provides advanced in-system programming and test capabilities for high performance, general purpose logic integration. Advanced features include output slew rate control and user-programmable ground pins to help reduce system noise. I/Os may be configured for 3.3V or 5V operation. All outputs provide 24mA drive capacity.

XC9500 device architecture consists of multiple Function Blocks (FBs) and IOBs fully interconnected by the FastCONNECT switch matrix. The IOB provides buffering for device inputs and outputs. Each FB provides programmable logic capability of 36 inputs and 18 outputs. The FastCONNECT switch matrix connects all FB outputs and input signals to the FB inputs. XC9500 family includes XC9536, XC9572, XC95108, XC95216, XC95288. Logic density of the XC9500 ranges from 800 to over 6400 usable gates with 36 to 288 registers [43].

Target chips

The programmability and adaptability have given FPGAs and CPLDs significant advantages for the development of prototype systems and their early introduction to the market. For these reasons the proposed controllers are to be implemented using these chips. A CPLD (XC95108-PC84, 2400 gates) device is used for implementing the chopper controller and a FPGA device (XC4010XL-PC84, 10,000 gates) is used for implementing the fuzzy logic variable speed controller.

8.1.2. The VHDL language for hardware modelling

With the conflict between control system complexity and the need for short and efficient design cycles, the use of high-level design languages is an attractive alternative to the schematic entry of traditional CAD. Although intended for digital hardware design, VHDL contains features found in modern programming language. The language provides the capability for developing models of systems that can be simulated prior to development and fabrication. This allows problems associated with integration, design, and even specification to be found early in the design cycle. This type of modeling also provides a mechanism for validating test plans and procedures prior to real hardware and software integration.

Hardware modelling by means of HDLs has changed the way designers think of circuits. Schematic entry tools have been replaced by CAD systems that support HDL specifications and synthesis. Making hardware design more closely allied to software programming. The change from gate-level to architectural-level modelling can be paralleled to the change from programming in assembly code to high-level software languages. Object-oriented methods for hardware specification have been proposed, and the unifying trend between hardware and software design is unavoidable, especially when considering that system design requires specifying both hardware and software components. The use of hardware description languages in the design process implies a different approach to design than that used in the past.

VHDL divides entities (components, circuits, or systems) into an external or visible part (entity name and connections) and an internal or hidden part (entity algorithm and implementation). An entity is defined, relative to other entities, by its connections and behaviour. After defining the external interface to an entity, other entities can use the specified entity when they are being developed. This concept of internal and external views is very important in the view of VHDL system design.

Generally, the main body of VHDL code describes the entity function involving some numerical calculations. The main numerical calculations involved in control systems are basic arithmetic operations such as addition, subtraction, multiplication and

division. These functions can be achieved in CPLD/FPGA with the VHDL description. The VHDL code describes digital logic circuits in a similar way to Boolean equations that represent logic functions. For example, in the add function, the Boolean equations that represent a 1-bit full-adder are:

$$Sum = ABC_{in} + \overline{A}\overline{B}C_{in} + A\overline{B}\overline{C}_{in} + \overline{A}B\overline{C}_{in} \quad (8.1)$$

$$Carry = AB + AC_{in} + BC_{in} \quad (8.2)$$

With VHDL code, if A and B are the two operands and sum and carry are the results of the 1-bit add operation, the function can be easily achieved as expressed below:

sum := A xor B xor $Carry_{in}$;

carry := (A and B) or (A and $Carry_{in}$) or (B and $Carry_{in}$);

The logic gates circuitry for the 1-bit adder is shown in Figure 8.2.

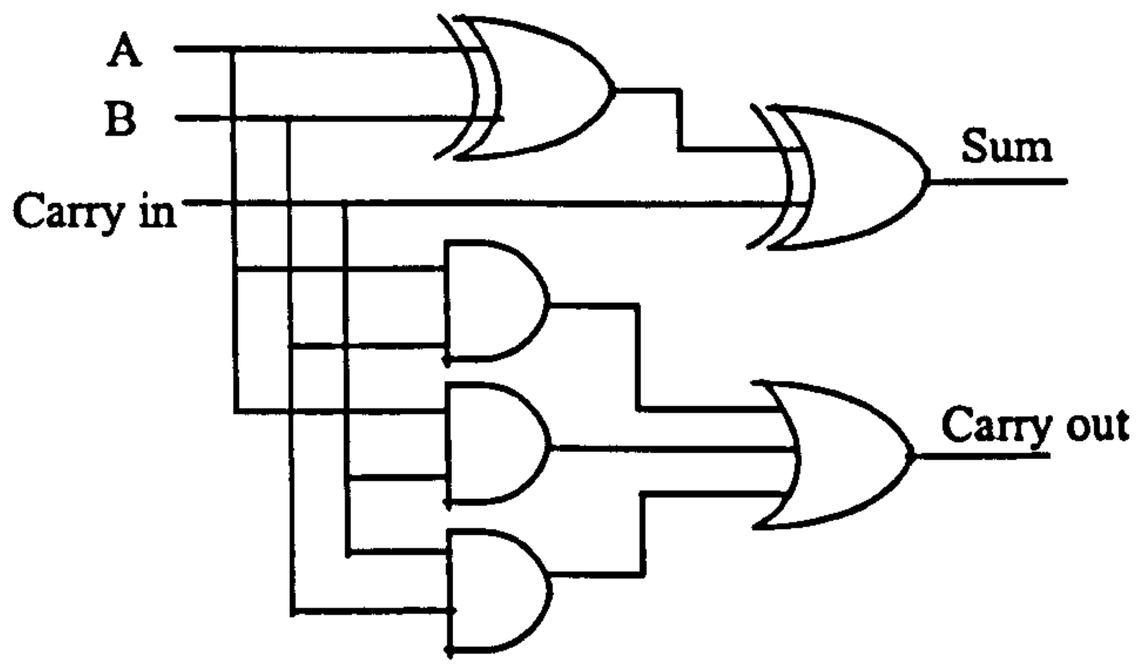


Figure 8.2. The illustration of the 1-bit adder VHDL code.

The VHDL code for a n-bit adder is:

```
carry := carry_in;
for i in 0 to n loop
    carry0 := carry;
    sum(i) := A(i) xor B(i) xor carry0;
    carry := (A(i) and B(i)) or
             (A(i) and carry0) or
             (B(i) and carry0);
end loop;
```

The for-loop and bit-to-bit operation construct the n-bits add operation, which can be illustrated in Figure 8.3.

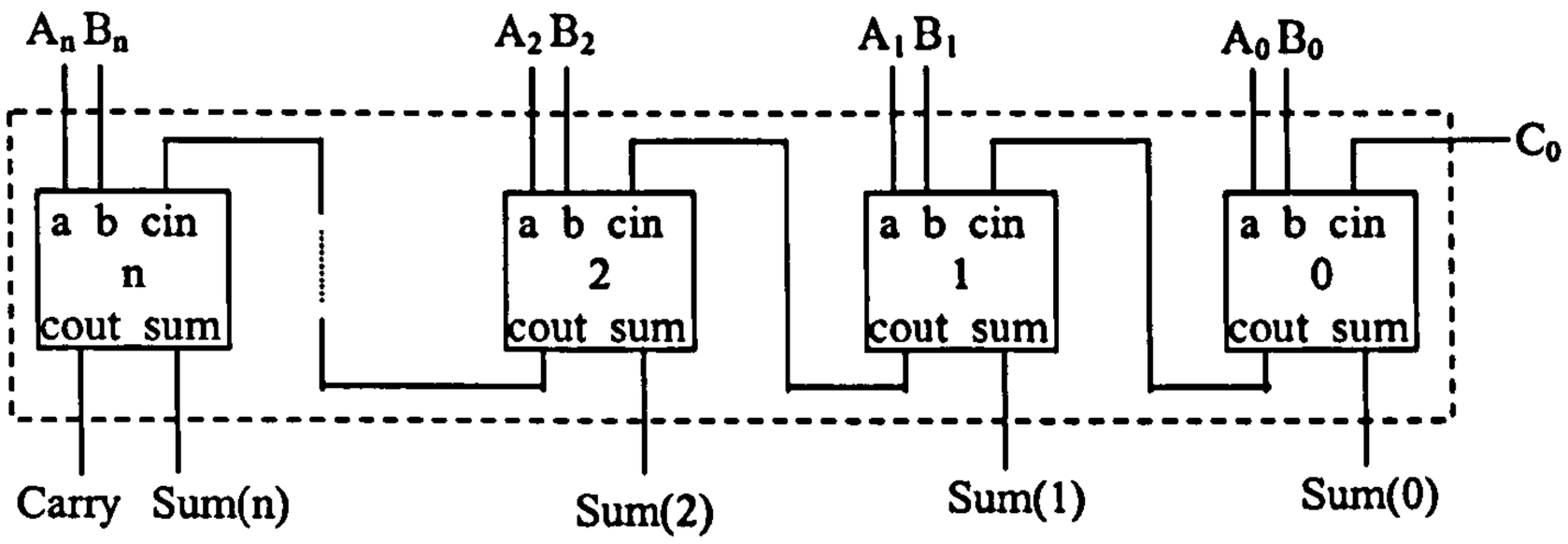


Figure 8.3. The illustration for a n-bit adder VHDL code

The above shows that the VHDL description is very similar to high level language but can describe the operation of digital electronics in a straightforward manner.

8.1.3. A system design tool -Foundation Express

Xilinx Foundation Express is a design environment for using VHDL to develop electronic systems based on FPGAs and CPLDs. The software package contains several productivity tools that are easily accessible through graphical tool bars, icons and pop-up menus. It supports hierarchically structured designs and can invoke the implementation programs as required to compile a design into a CPLD/FPGA. It supports the complete spectrum of programmable logic design methodology from fully automatic to hand-crafted. All the tools feature a Graphical User Interface (GUI). On PCs, the GUI is fully Microsoft Windows compliant, all programs are executed from tool bars and icons. The main productivity tools are briefly introduced as follows.

The *Design Manager* provides a complete project management environment for a wide range of device families. It provides version control, device re-targeting and design reuse. The implementation process can be controlled with the configurable *Flow Engine*. The system can be analysed and optimised before the implementation. *Floorplanner* provides techniques that have proven to be extremely valuable to gate array and custom silicon designers. It allows optimisation of specialised structures like high-speed distributed RAM and three-state internal bus features. The interactive *Timing Analyzer* makes it easy to quickly determine the performance of the designed system by generating custom timing reports. Using pop-up menus, the delay along a

specific path or group of paths can be quickly shown. The delay along all paths of a certain type or those associated with a specific clock signal can also be shown. In addition, the *Timing Analyzer* automatically displays the estimated maximum frequency allowed for the design. The *Hardware Debugger* allows the verification of configuration data and the internal signal activity to be viewed. It takes advantage of the reprogram ability of SRAM-based devices by configuring the chip in-circuit using a cable connected to the computer.

Design using Xilinx Foundation Express consists of three interrelated steps: entry, implementation and verification.

Design Entry. Foundation Express design entry provides both VHDL editor and schematic editor, which makes visually oriented design techniques compatible with text-based VHDL design constructs. The design environment supports hierarchical design entry, with top-level drawings defining the major functional blocks, and lower-level descriptions defining the logic in each block. Different hierarchical elements can be specified with different design entry tools, allowing the use of the most convenient entry method for each portion of the design.

Design Implementation. After the design is entered, the hierarchical elements are combined using the implementation tools. Foundation Express translates VHDL to an internal format and synthesizes VHDL descriptions according to the VHDL synthesis process defined. Then implementation tools can be used to map the logic into the resources of the target device architecture, determine an optimal placement of the logic, and select the routing channels that connect the logic and I/O blocks so as to create the CPLD/FPGA configuration.

Design Verification. Design verification typically involves a combination of in-circuit testing, simulation, and static timing analysis. The user-programmable nature of these devices allows designs to be tested immediately in the target application by direct downloading the Serial Vector Format (SVF)/bitstream file from a computer to a CPLD/FPGA device. Static timing analyses can be performed to examine the logic

and timing of the implementation through calculating the performance along signal paths, identifying possible race conditions and detecting set-up and hold-time violations.

8.1.4. The design procedure

As previously mentioned, VHDL has many advantages, however, VHDL by itself cannot provide all of these benefits. Therefore an organised design methodology must be developed to accommodate modelling activities, along with a design framework in which the system models are created. This framework implies an efficient method of VHDL modelling and simulations activities, and a technique for evaluating simulation results.

The stages involved in designing an electronic system are illustrated in Figure 8.4, which shows the complete electronic design process, from the specification of the system, through hardware and software partition, down to the specification and implementation of the hardware and software parts of the completed system.

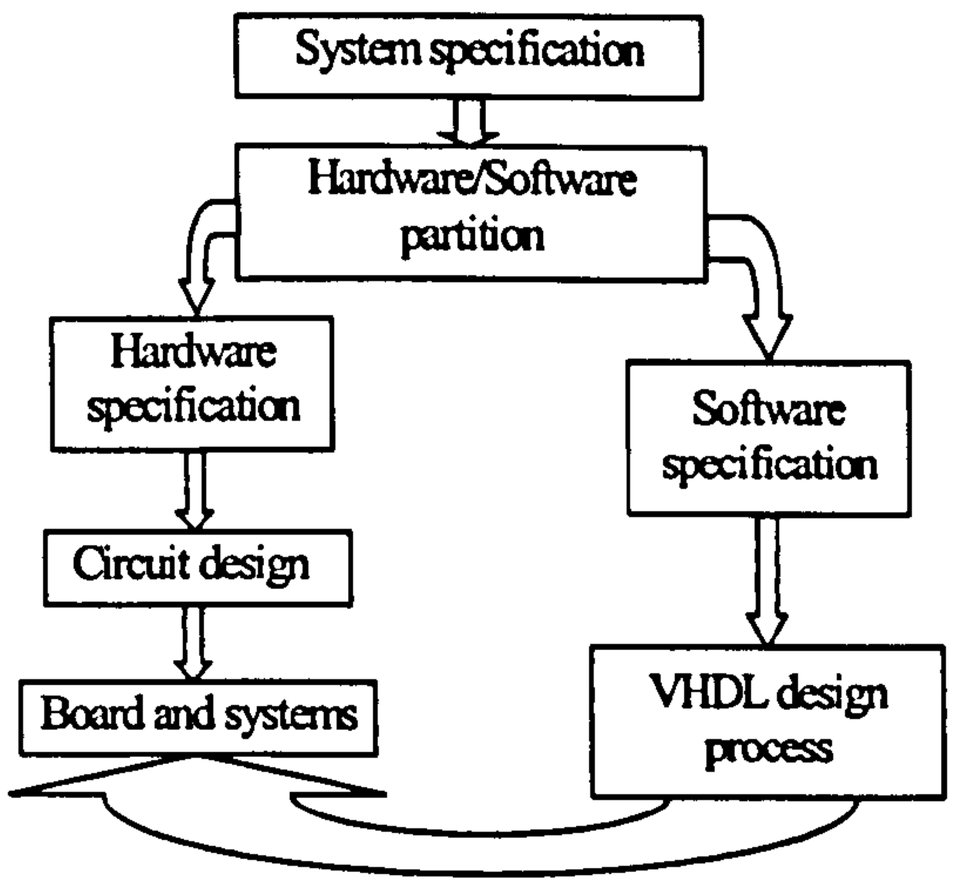


Figure 8.4. The stages of electronic system design

The hardware includes the required analogue circuit and the interface to the CPLD/FPGA. The software is used to describe the CPLD/FPGA structure and perform the required function.

Figure 8.5 shows the design process of using VHDL.

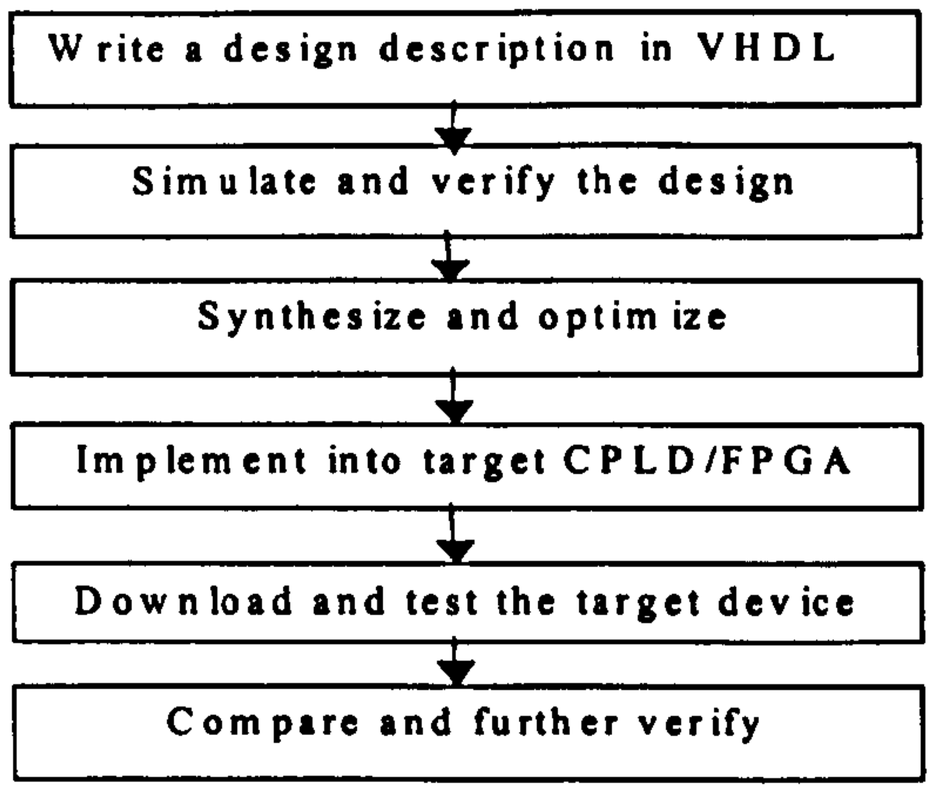


Figure 8.5. Design Flow

The above design steps are.

- Write a design description in VHDL. The description can be a combination of structural and functional elements.
- Simulate and verify the design. Simulate the design by using the Foundation Express simulator and verify whether the description is correct.
- Synthesize and optimize. The VHDL design descriptions are synthesised and optimised into a gate-level netlist. Foundation Express synthesize tool generates the optimised netlists for the targeted CPLD/FPGA architecture.
- Implement into target CPLD/FPGA. Which is a translation and optimization process to forward annotates timing constraints to the place and route engine so as to produce serial vector format (SVF) file/bitstream file.
- Download and test the target device. The SVF/bitstream file is downloaded for CPLD/FPGA configuration. This step is to link the CPLD/FPGA technology-specific version of the design to the development system. Two development systems for the targeting devices are used, they are XC4010XL test board and XC9500 test board (More details about the development boards are described in Appendix A).
- Compare and further verify. Compare the output of the gate-level simulation against the output of the original VHDL description simulation to verify that the implementation is correct.

All these steps are performed with the Xilinx Foundation Express tool. The Xilinx working environment provides a fully automatic design system to perform various functions, such as the simulation, synthesize, optimise, implementation, download and configuration the target devices.

8.2. The control system

8.2.1. Control system structure

The structure of the control system implemented with a FPGA (XC4000XL-4010XL - PC84) and a CPLD (XC95108-PC84) is shown in Figure 8.6.

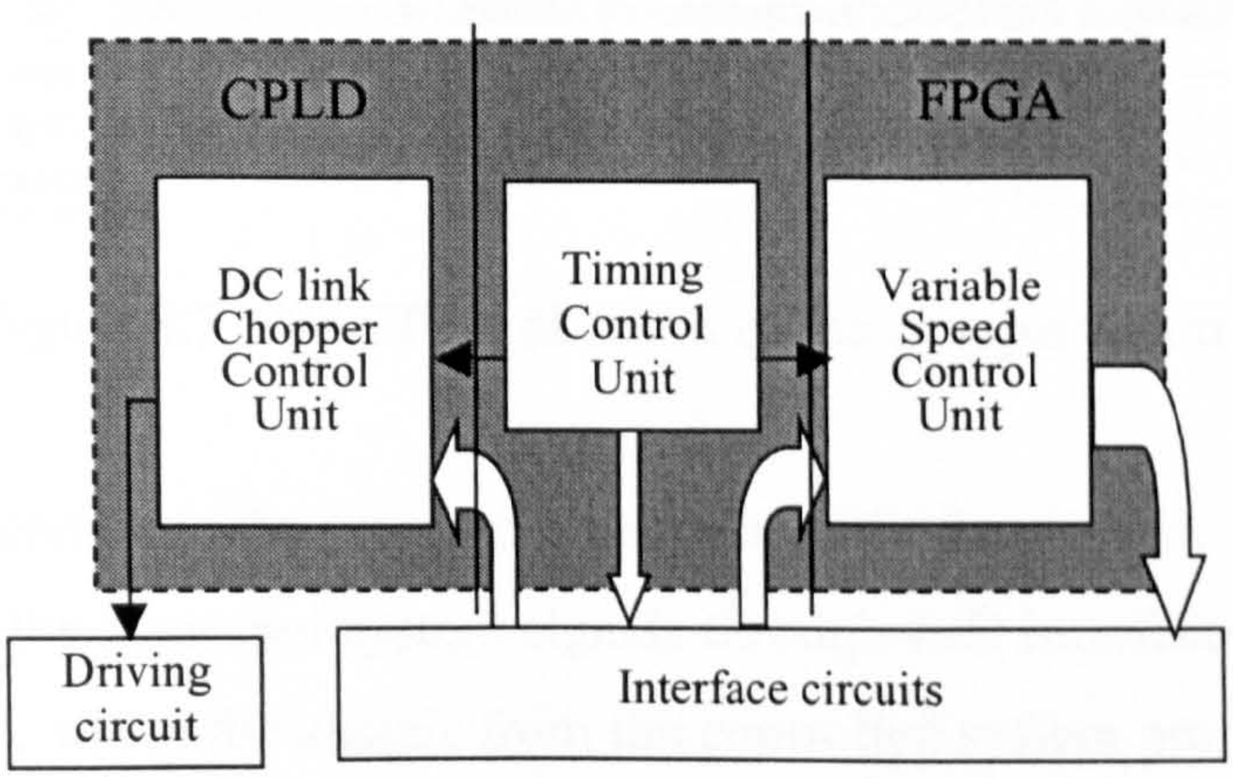


Figure 8.6. FPGA and CPLD control system

The control system consists of two main ingredients: chopper control unit and variable speed control unit. The timing control unit generates a timing signal providing the on-chip timing signal. This process has two main advantages: (1) flexibility: The actions of the control system are controlled by internal timing signal that is embedded in the chip; (2) simplicity: reducing hardware circuitry complexity. The two main units are co-operative controllers. One is for the dc link voltage control, and the other one is for the variable speed control. In the following sections the design phases of utilising VHDL, including VHDL modelling tasks, logic synthesis, integration tasks, and final end-item test, are described.

8.2.2. CPLD implementation of chopper controller

CPLD skeleton of the chopper controller

The structure of this controller is shown in Figure 8.7. The CPLD functions of the controller are covered by a dashed line frame.

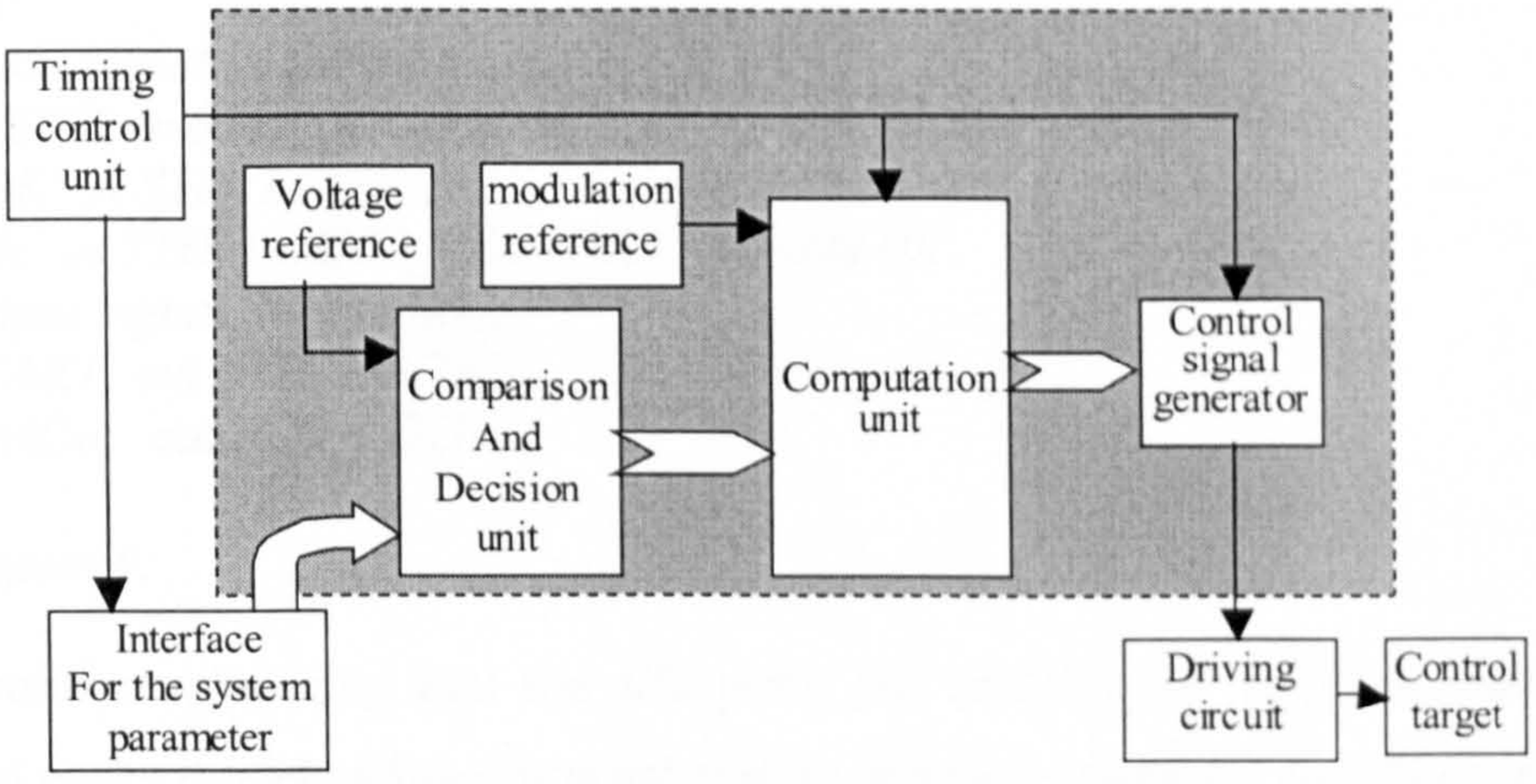


Figure 8.7. The CPLD skeleton of the chopper controller

The voltage reference is the required voltage level in the system. The digital signals converted from the measured system signals through A/D interface and then enter the comparison unit, where the signals from the controlled system are compared with the reference to determine whether the system voltage is higher or lower than the required voltage level. According to this information, the computation unit decides the adjustment of the control signal, t_{on}. Then the control signal generator provides the control output that is sent to the dc-dc converter through the driving circuit. The control system is implemented as a real time feedback control loop. The timing control unit controls the sequence of each stage in the closed control loop.

Chopper controller entity modelling

Based on the system requirements and validation from system simulations in Chapter 5, a viable hardware and software partition can be performed while the functionality of the system or subsystem is investigated. The dc chopper controller central unit is the target chip, XC9500-95108PC84, which has a number of input/output lines allowing

signals to transfer through I/O ports of the target chip, they are the communication link between the controller entity and the external system.

The VHDL descriptions for the chopper controller entity I/O ports are as follows:

```
entity chopsim9 is
   port (
   -- input signal declarations.
      RESET: in STD_LOGIC;
      CLK: in STD_LOGIC;
      Vdc: in STD_LOGIC_VECTOR (7 downto 0);
   -- output signal declarations.
      START: out STD_LOGIC;
      CtrlOut: out STD_LOGIC
   );
end chopsim9;
```

It is worth emphasising that the I/O ports are actually the interface between the required entity model and the external system, through which the data flow in and out. The functional behaviour of the chopper controller can be examined without detailed bit level interactions.

The chopper controller operates in a closed loop control system, whose function is to produce a chopper switching control signal for dc link voltage control. The dc link voltage (denoted as Vdc) is the input signal and the switching control signal (denoted as Ctrlout) is the output signal of the controller. The input signal Vdc is defined as a 8-bit vector to match an 8-bit A/D converter. The output signal is defined as a standard bit signal, which is used as the driving signal to control the dc link chopper "ON" or "OFF".

Inside the chopper controller entity, the architecture constructed with the functions shown in Figure 8.7 are achieved by interconnection of the programmable logic devices inside the chip. The architecture body carries out the comparison, calculation, adjustment decision and control signal generation. The flow chart is shown in the figure 8.8.

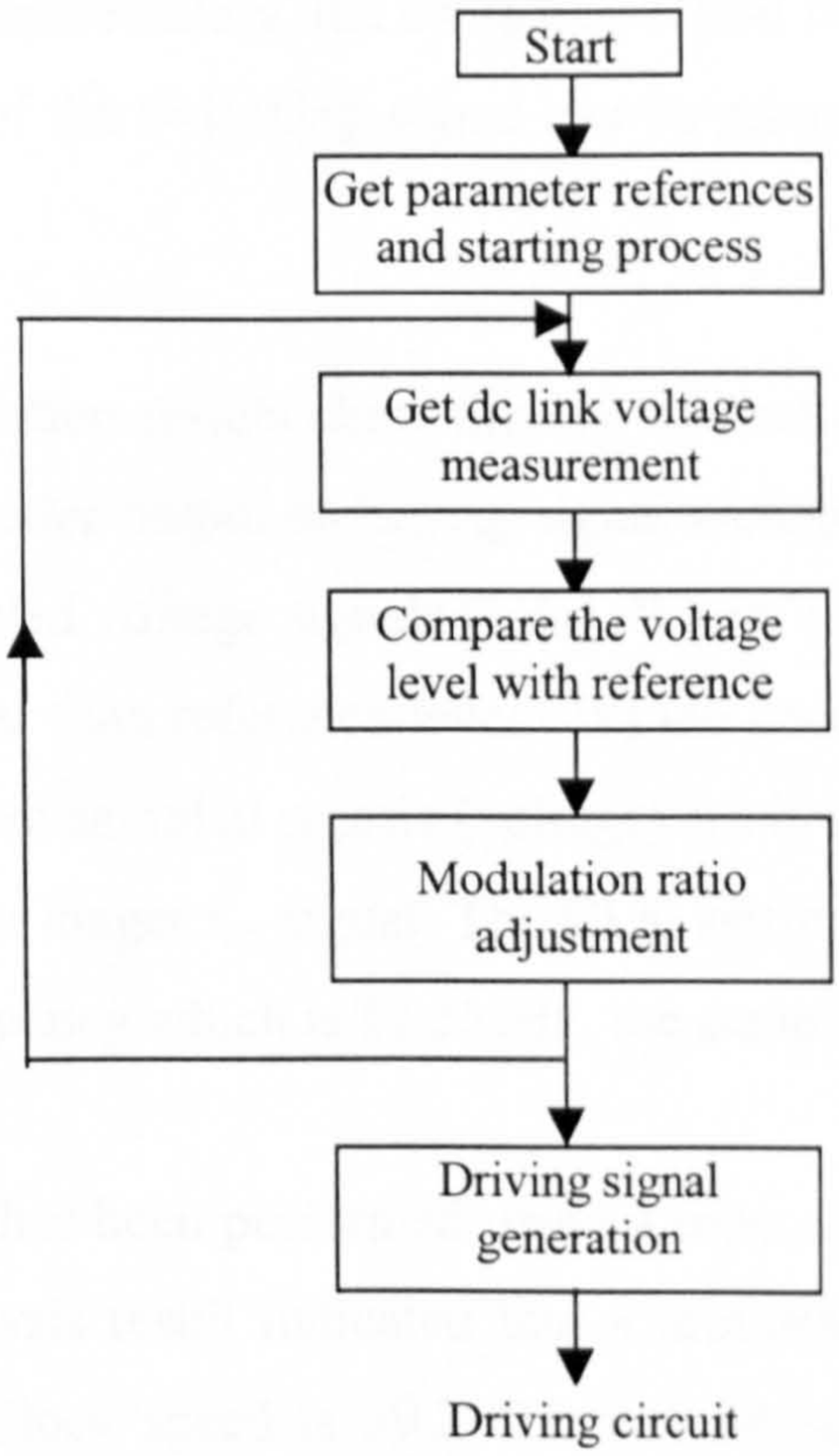


Figure 8.8. Flow chart of the chopper controller

Functionality test and timing analysis

To verify the performance of the chopper control system, the VHDL modelled chopper controller has been synthesised and simulated in both functionality and timing suitability. Figure 8.9 shows the controller's functional simulation results.

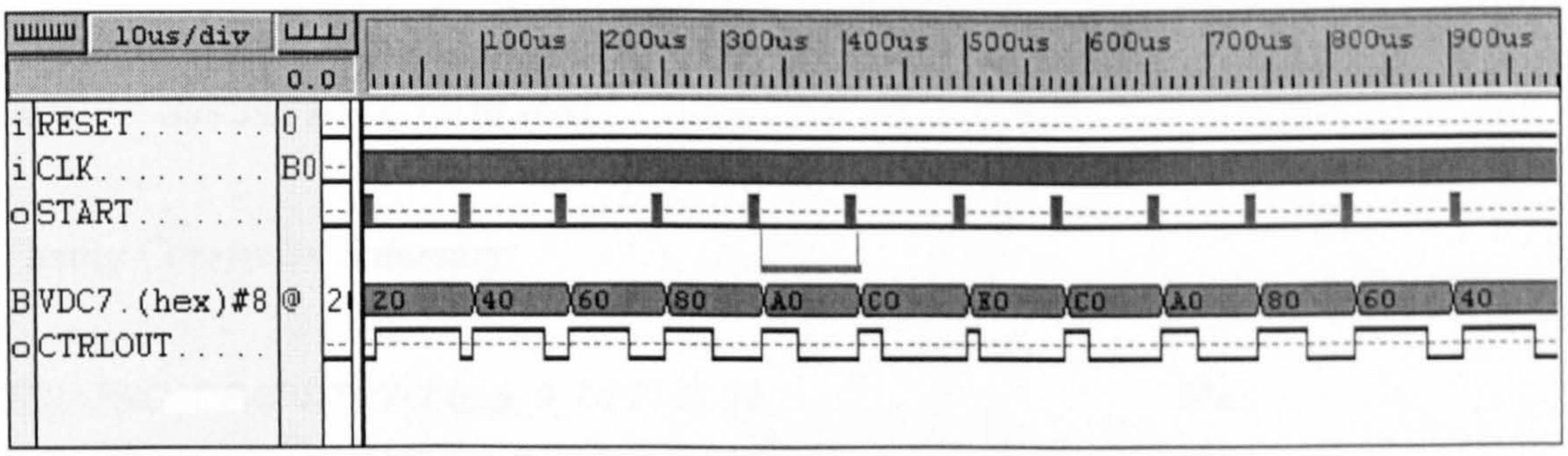


Figure 8.9. Functionality simulation results of the chopper controller in VHDL

It is expected that when the dc link voltage level becomes higher or lower than the reference value (for this simulation the reference setting is 7F) the output of the

controller responds immediately, the switching signal becomes narrowed or widened. The ON-state ratio of the switching signal can be programmed to vary in the desired range.

The functional simulation results show that the controller has correctly performed. In Figure 8.9, the controller output switching signal changes its width in response to the variation of the sampled voltage signals (Vdc). When Vdc (represented with two bytes HEX code) are higher than reference level (7F) the controller responds with a shorter t_{on} output signal. When sampled signals (voltage) are lower than the reference level the controller generates a longer t_{on} signal. The CLK setting for the functional simulation is at basic clock frequency which is 12.5MHz, the sample interval is 82.08us.

The timing analysis has been performed using a time simulation with 12.5MHz clock frequency. The analysis result indicated that a minimum clock period is 25.5ns and maximum internal Clock speed is 39.2MHz can be used for the chopper controller CPLD implementation. The following is part of the analysis report generated from the simulation.

Performance Summary Report

Design: fdown

Device: XC95108-7-PC84

Program: Timing Report Generator: version M1.5.25

Date: Sun Jun 10 14:54:10 2001

Timing Constraint Summary:

TS0=FROM:PADS(*):TO:tgrp_0_DFF:83.00 *Met*

TS1=FROM:tgrp_0_DFF:TO:PADS(*):83.00 *Met*

TS2=FROM:tgrp_0_DFF:TO:tgrp_0_DFF:83.00 *Met*

Performance Summary:

Clock net 'CLK' path delays:

Clock Pad to Output Pad (tCO) : 4.5ns (1 macrocell levels)

Clock to Setup (tCYC) : 25.5ns (2 macrocell levels)

Setup to Clock at the Pad (tSU) : 5.5ns (0 macrocell levels)

Minimum Clock Period: 25.5ns

Maximum Internal Clock Speed: 39.2Mhz

(Limited by Cycle Time)

8.2.3. Hybrid variable speed controller FPGA implementation

Simplicity and fewer requirements for intensive mathematical computation are the most important advantages of fuzzy logic method over most other control techniques. This feature allows the fuzzy logic controller to be easily implemented using inexpensive hardware technology. The low-cost FPGA technology (with XC4000XL-4010XL-PC84) has been used to implement the fuzzy logic core based hybrid variable speed controller.

The hybrid variable speed controller is modeled and implemented using the hardware description language (VHDL). Despite the simplicity of the fuzzy logic core algorithm, it is still a challenge task to implement the controller in a single FPGA chip within the gate limitation. The reason is that the realization of all the mathematical operations in an FPGA consumes a lager number of gates. However, fuzzy logic can tolerate some ambiguities and therefore fortunately it does not require high calculation accuracy. Consequently some simplifications and approximations can be tolerated without reducing the performance of the controller.

FPGA skeleton of the hybrid variable speed controller

The final form of the hybrid variable speed control system includes the variable speed reference decision, speed deviation measurement algorithm, the fuzzy logic core (FLC) algorithm, and the interface between the controller and the controlled system. The FLC input, fuzzification, rule inference, and defuzzification modules were

integrated together to form the structure of the FLC. Figure 8.10 shows the block diagram of the FPGA based variable speed controller. The grey area shows the FPGA functions of the controller.

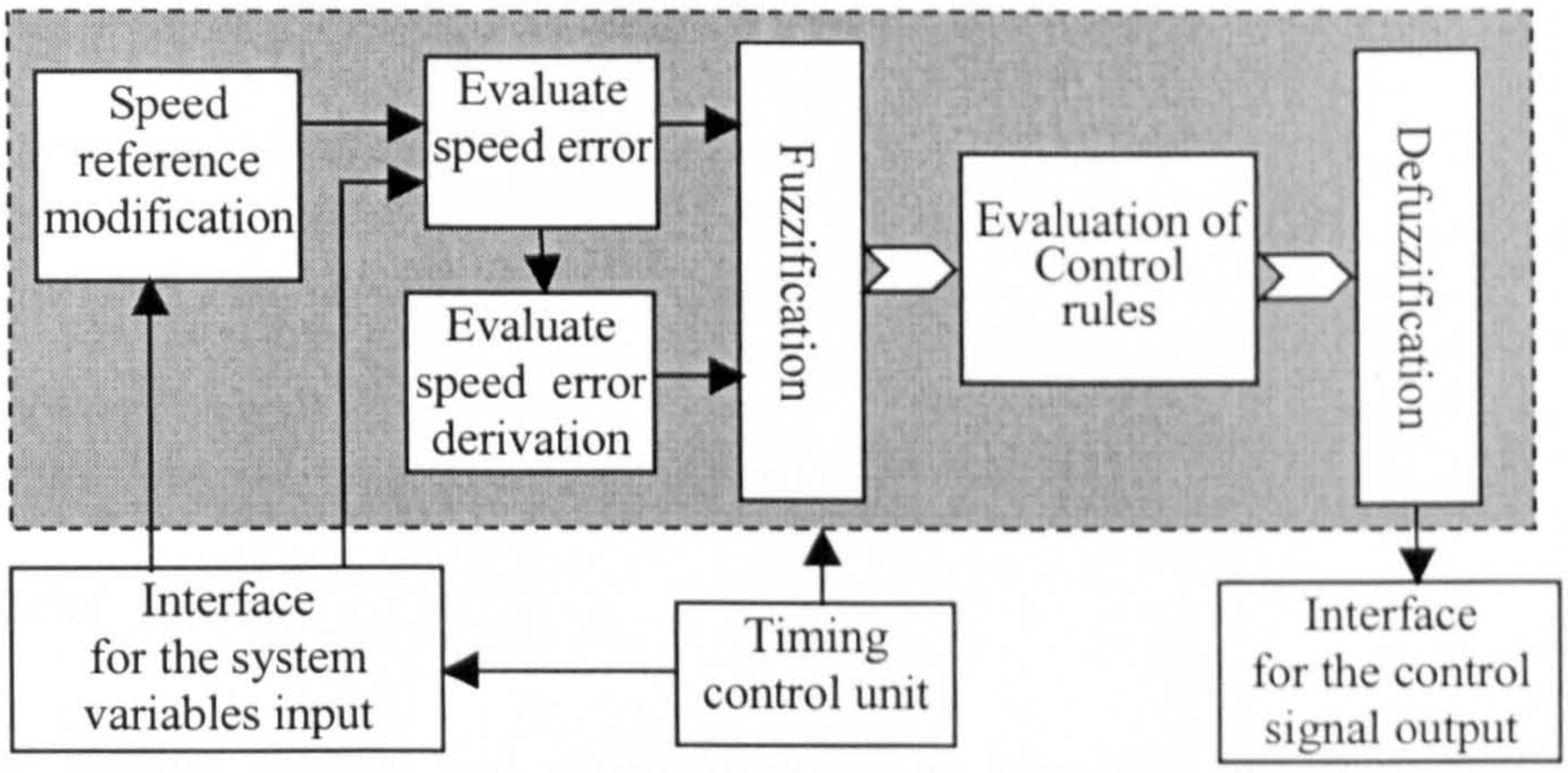


Figure 8.10. Block diagram of FPGA based controller

In the diagram, the speed reference is modified according to system load conditions. Once the reference speed is modified, the error between the reference and actual speed and the derivative of the error are calculated to produce the information required by the fuzzy logic core to control the fuel input and therefore speed. In this closed loop control system, the input signals are the system current and machine speed, the output is the signal to the actuator. In practice, for FPGA application, the measured system signals (current and speed) need to be converted into digital signals for the inputs of the FPGA based controller, then the digital output of the controller needs to be converted back to analogue form. Therefore A/D and D/A interfaces are needed for the FPGA variable speed controller

Hybrid variable speed controller entity modelling

Use a similar method to that used for CPLD chopper controller, the variable speed controller is modelled with the specified system requirements and simulated to verify the control strategy. The entity definition and the associated port declarations are:

entity wholectrl_FH2test_L12R is

```
port (
        RST: in STD_LOGIC;
        CLK: in STD_LOGIC;

        --inputs
        Speed: in std_logic_vector(7 downto 0);
        Pcur: in STD_LOGIC_VECTOR (7 downto 0);

         --operating condition
        sign: out std_logic;
        abnormal_sig: out std_logic;

        --control output
        Uout1: out std_logic_vector(7 downto 0)
);
end wholectrl_FH2test_L12R;
```

Two input signals, current and speed (denoted as Pcur and Speed respectively), are measured from the system and then converted into digital signals through A/D interfaces. In the controller, the two input parameters, Pcur and Speed, are defined as two 8-bit vectors to match 8-bit A/D converters. There is one output signal, Uout1, which is the control signal to the actuator. The output signal is defined as an 8-bit vector signal to match an 8-bit D/A converter. There are a few other signals listed in the port declarations. They are for monitoring system operating conditions, typically, the abnormal signal indicating whether the system is overloaded or the running speed is too high.

The architecture of the VHDL code describes the function of each part in the controller, including the following: calculation of the reference speed; calculation of the speed error and the derivation of the speed error; fuzzy reasoning to produce the adjustment command for fuel control; fuzzifycation and defuzzification. These functions are achieved in FPGA with VHDL description. The flow chart for the fuzzy logic core based hybrid variable speed controller is shown in Figure 8.11.

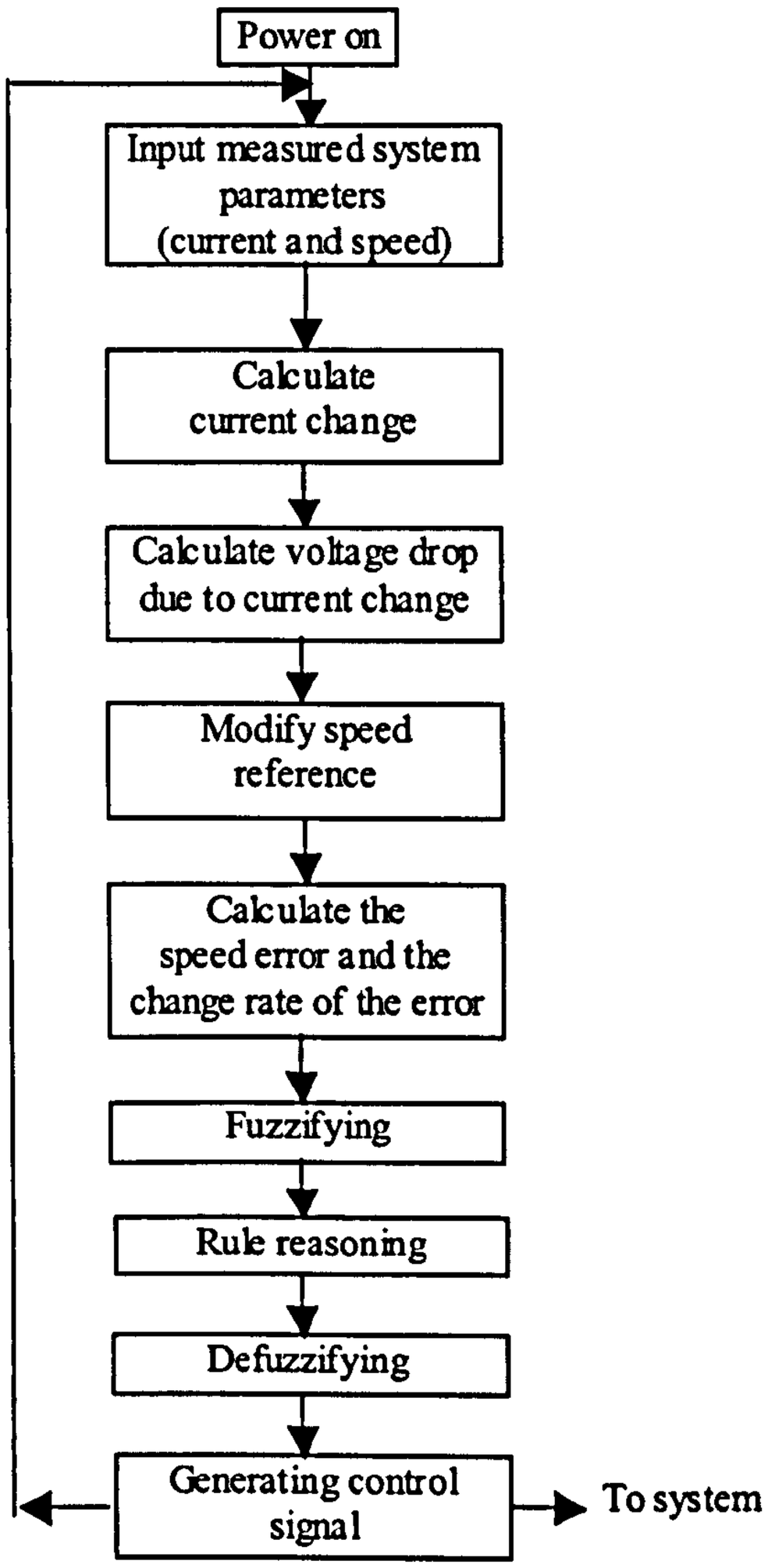


Figure 8.11. Flow chart of the variable speed controller

Variable speed controller functionality test and timing analysis

The functional simulation results of the variable speed controller are shown in Figure 8.12 and Figure 8.13. The CLK setting for the functional simulations is 200MHz.

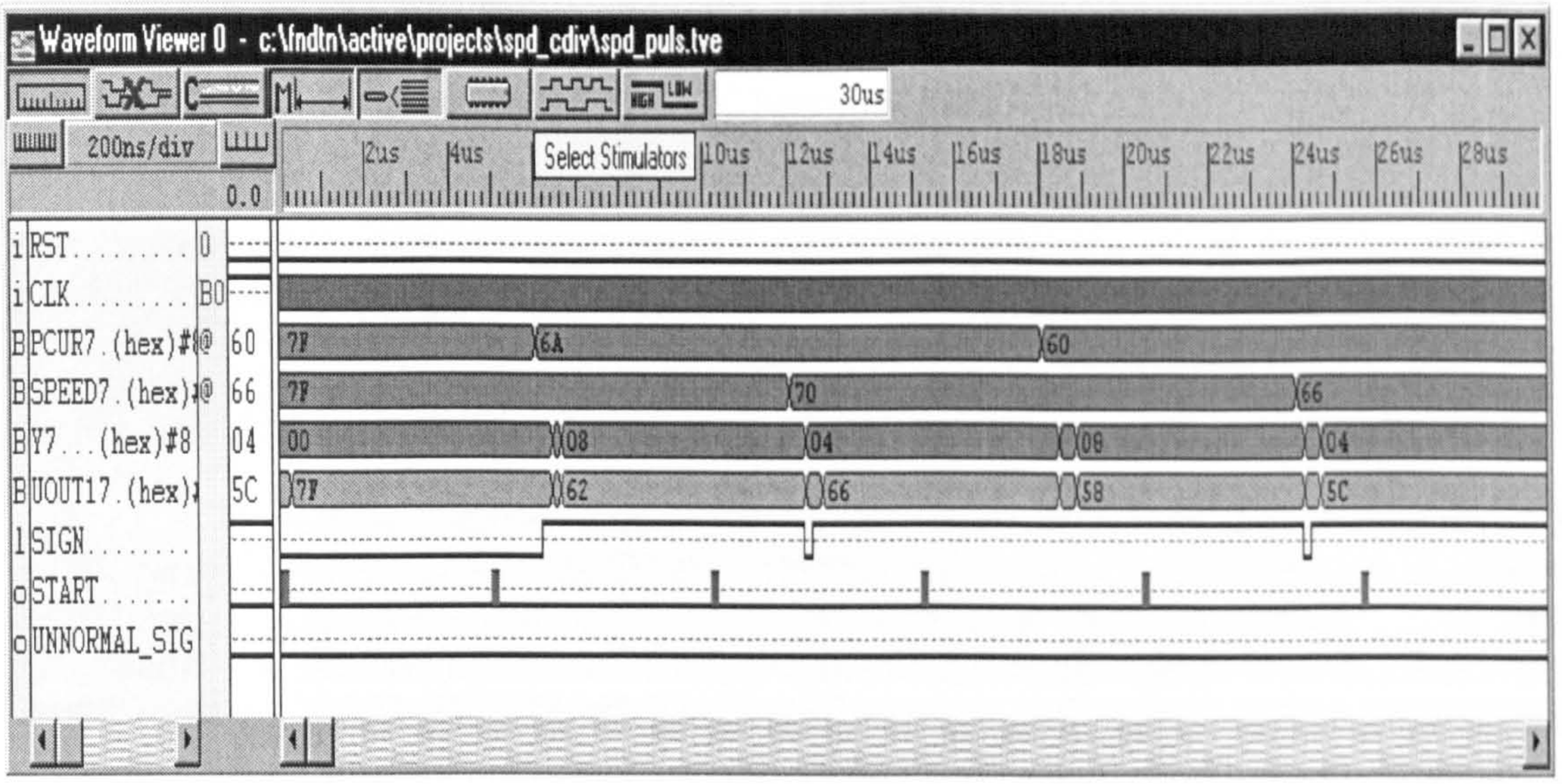


Figure 8.12. Functional simulation results of variable speed controller model in VHDL (load decreasing)

Figure 8.12 shows the cases of load (represented by *Pcur*) decreasing. Assuming the system is running at a normal condition at t=0, the signals, *Pcur, speed* and the control output *Uout1* are all 7F. Then the signal of *Pcur* is suddenly changed from 7F to 6A, representing a load decrease. Due to this load change the control signal changes from 7F to 62, indicating there is a control output adjustment corresponding to the load decrease. For the stand-alone generating system, the lower level of control output means less fuel input. The amount of the decrease is decided by considering the load change magnitude as well as the speed error and the change rate of the speed error. The change of *Uout1* caused by speed and speed change was 08, the remainder being due to the decreased load. At t=7 us, the speed changes from 7F to 70 due to Uout1 change and because the speed change results in a reduced speed error, then *Uout1* changes from 62 to 66. During this period, the signal *sign* changes from low to high, which indicates that operating speed was lower than required, then overshot and became higher.

When the load again changes (t=18 us), *Uout1* reduced to 58. This output reduction corresponds to load shedding, the *sign* signal indicates that the speed is not higher than the required speed. This condition continues until t=25 us but the speed error reduces as seen in trace resulting in the control signal being 5C. Signal Y indicates the

affect of speed and speed change rate on the fuel input adjustment. The operating speed reaches a new steady state when Y becomes zero.

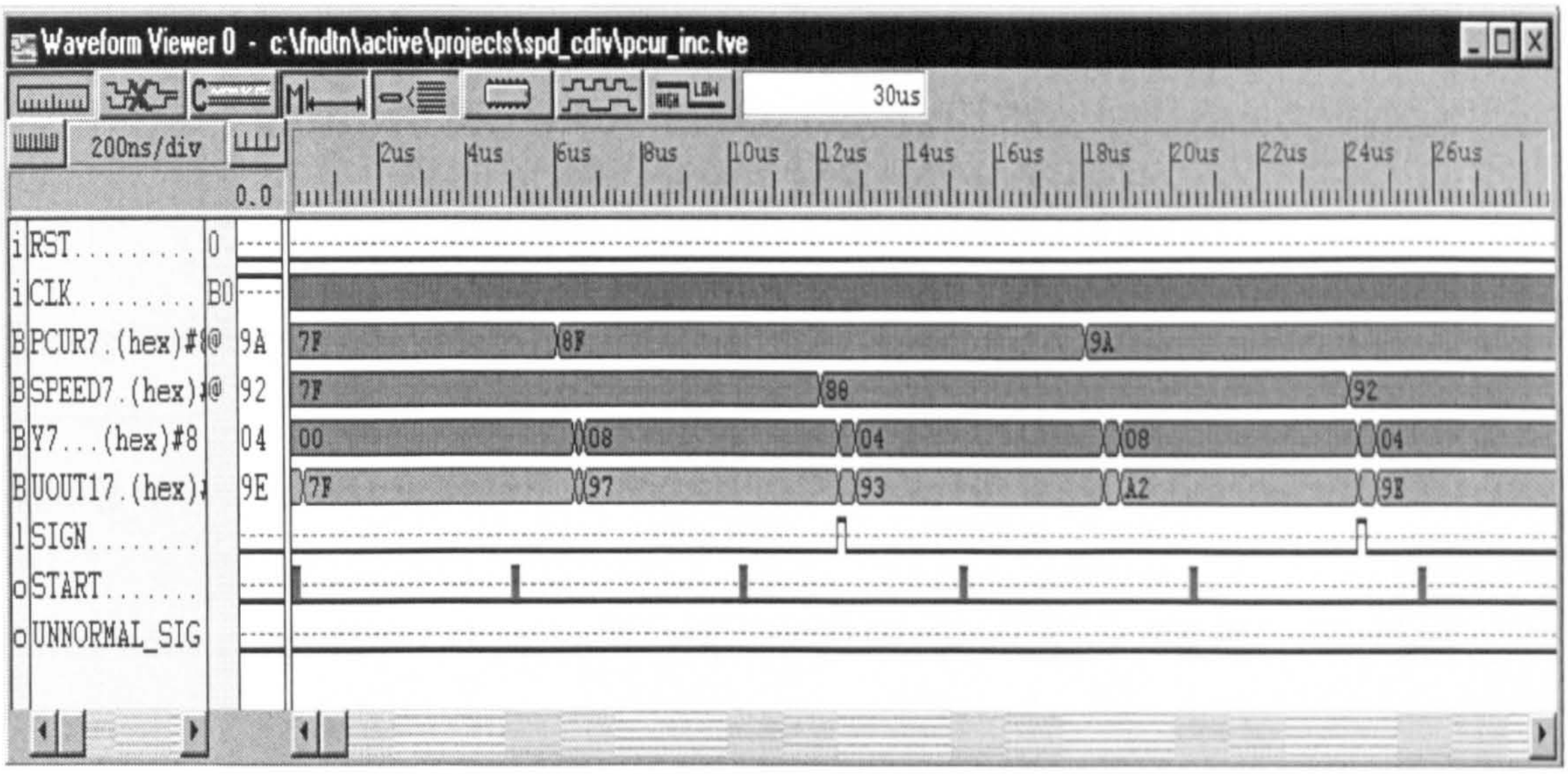


Figure 8.13. Functional simulation results of variable speed controller model in VHDL (load increasing)

Figure 8.13 shows the results for the case of load increasing. When the load changes from 7F to 8F and then 9A, the fuel control was adjusted from 7F to 93 then to 9E with the effects of speed change being included.

The results from both functional simulation cases show that the performance of the controller is functionally correct.

The timing analysis was performed at a clock frequency of 12.5MHz. The analysis result indicates that a clock signal with a minimum clock period of 63.434ns (maximum frequency 15.764MHz) can be used for this variable speed controller FPGA implementation and the maximum path delay from/to any node is 17.154ns. The following is part of the analysis report.

Timing analysis report

Xilinx TRACE, Version C.16

Design file: *spd_cdiv.ncd*

Physical constraint file: *spd_cdiv.pcf*

Device,speed: *xcs40,-4 (x1_0.14.2.2 1.7 PRELIMINARY)*

Timing summary:

Timing errors: 0 Score: 0

Constraints cover 1568133 paths, 0 nets, and 3236 connections (96.7% coverage)

Design statistics:

Minimum period: 63.434ns (Maximum frequency: 15.764MHz)

Maximum path delay from/to any node: 17.154ns

Minimum input arrival time before clock: 61.204ns

Minimum output required time after clock: 16.046ns

Analysis completed Mon Oct 09 16:16:05 2000

8.3. Summary

The development of the controllers was carried out using the Xilinx development system, which provides programmable logic specific high-level flows and optional auto-interactive tools. In order to reduce the size and cost of the control system, the controllers are implemented in XC9500 and XS4010 devices which are compact and cheap.

With top-down design methodology, the design started at a higher-level of abstraction than traditional gate-level design techniques. The logical or functional abstractions are

manipulated using logic synthesis tools for the CPLD/FPGA implementation. Typical functionality tests have been performed with the simulation tool to verify the CPLD/FPGA design in the form of a VHDL description at the early stage in the design process. The Foundation Express simulator is used to verify that the description, test architectural and design decisions and functionally correct at a high level before implementation is performed at gate level. Static timing analyses are also performed to examine the logic and timing of the implementation and calculate the performance along signal paths, identify possible race conditions, detect set-up and hold-time violations, which decide the timing suitability for prototyping.

Foundation Express, the environment for VHDL development, provides computer added design process technology so that the logic optimisation can automatically transform a synthesised design to an efficient circuit (the CPLD/FPGA prototyping). Practical experiments using the controllers developed are presented in the next chapter.

Chapter 9

Experimental Studies

This chapter presents the experimental studies carried out to validate the operation of the proposed systems. The implementation of the hardware model configurations is described and the implementations of the FPGA and CPLD based control systems are presented. The performance of the proposed control systems is tested and results are presented.

9.1. Electrical machine and power electronic test system

The simulation results given in Chapter 7 have demonstrated the effectiveness of the chopper controller and the hybrid variable speed controller. The objective of the experimental work is to validate the simulation results. A diesel engine and PM generator set could ideally fulfil the task, however, at the time of carrying on the research work, such equipment was not available in the department, therefore, an electrical machine experimental system was set up. The experimental system includes:

- a FH2 MKIV electrical machine system shown in Figure 9.1, including a dc motor and an ac generator with fixed excitation, is used to simulate the diesel engine and generator system
- a power electronic system to perform power conversion
- control systems to control the prime mover and power electronic converters
- an interface and measurement system.

The FH2 MKIV electrical machine system and power electronic conversion system are described in this section. The control systems and associated interfaces are discussed in following sections.

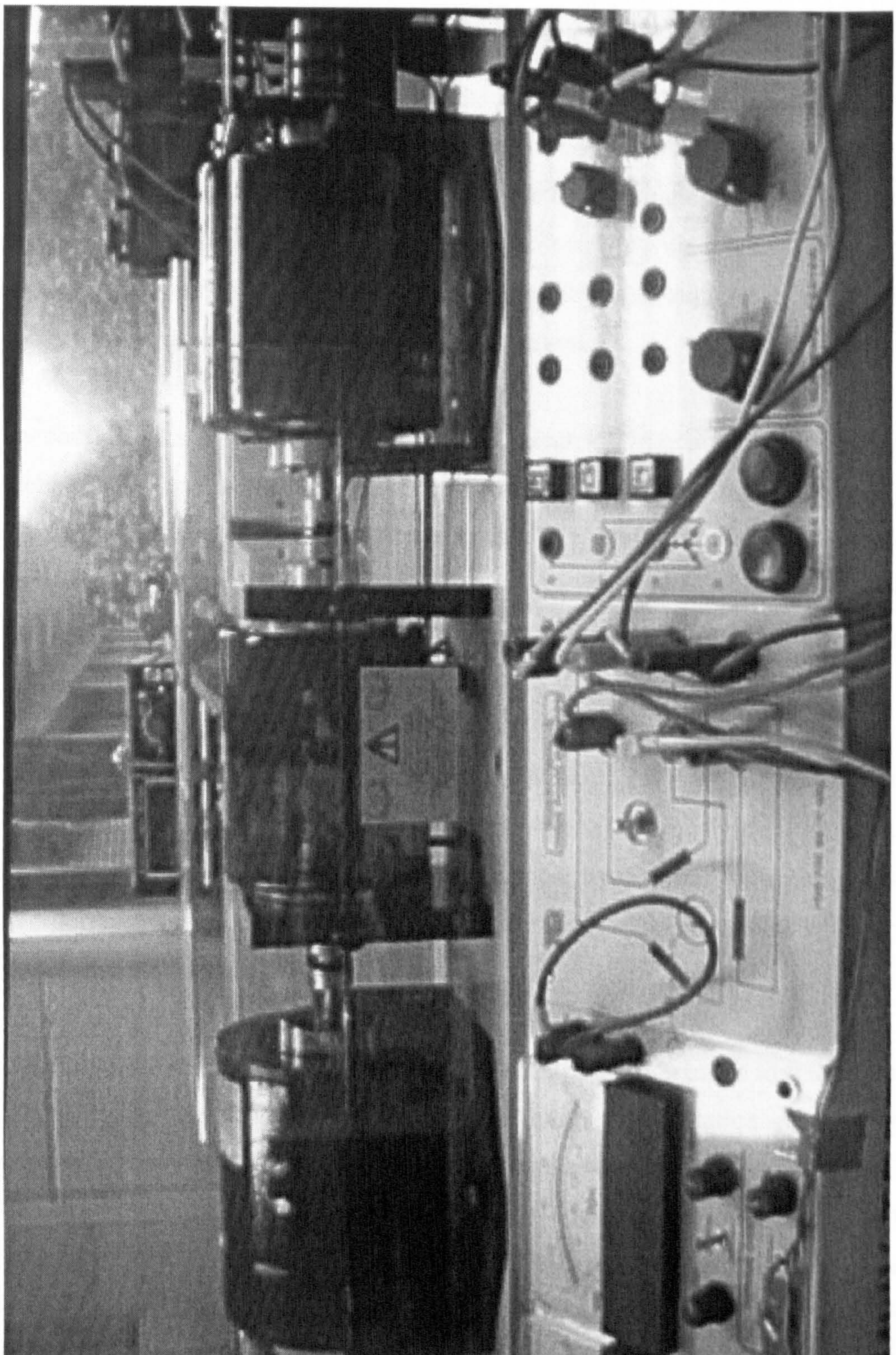


Figure 9.1. The FH2 test bed configuration

FH2 MKIV electrical machine system

The FH2 MKIV Electrical Machine Teaching System shown in Figure 9.1 is the latest version of the fractional horsepower machine teaching system designed for educational purposes. A series of ac and dc machines can be fitted into the system to conduct various electrical machine experiments. The FH2 MKIV test bed is fitted with a sliding clear Perspex safety guard covering the dynamometer and couplings in order to reduce the possibility of the user coming into contact with rotating parts. This unit consists of a rigid steel chassis on to which the basic power supplies, brake and control facilities and machine mounting cradles are mounted.

The configured test system consists of two electrical machines, a DC Compound Machine FH50 and a Wound Rotor Machine FH100. In the system, an optical sensor built into the frame of the dynamometer produces a series of pulses, the frequency of which is proportional to the speed of rotation. Electronic circuitry converts the frequency signal into a proportional dc voltage, which is sent to the control system as the speed measurement signal.

The DC machine (shunt connection, 100W, 110V and 1500 rev/min) placed on the left of the machine test cradle works as the prime mover. Obviously the fuel input to the diesel system is continuously adjustable, therefore to simulate the diesel engine and governor system in the experimental study, an equivalent electronically controllable prime mover is required.

This can be provided by the brake system, which can apply a brake torque to the drive, The level of brake torque is adjustable by varying the excitation current, which is supplied by a Brake Torque Control Potentiometer. The Brake Torque Control Potentiometer supplies voltage in the range of 0 to 30V dc and can be controlled by an analogue dc voltage signal. A 0-5V dc input signal is needed to generate the full output range of the supply, namely 0-30V. In this way the adjustable torque can be used to simulate the fuel input control of the diesel engine system. Maximum braking torque operation represents minimum fuel input and minimum braking torque operation represents the maximum fuel input. Therefore an increase of the fuel input is

effectively a decrease of the braking torque, and vice versa. In this way, the test system can be controlled by the signal produced by the controller implemented within the FPGA. Consequently, the performance of the developed control system can be fully tested and the control strategies validated.

The permanent magnet generator is simulated by a wound rotor ac electrical machine FH100 (4 poles, 80W, 230V, delta connected, 50Hz, at 1500 rev/min) which is operated as a synchronous generator with a fixed dc current supplied to the rotor winding. Loading resistors are used as load for the experimental study.

The power electronic system

The power electronic converter system includes a rectifier, a dc-dc converter and an inverter while a rectifier module is used to perform the ac to dc power conversion. The three-phase ac generated voltage is rectified by a diode rectifier and then connected to the input of the dc-dc converter. The output of the dc-dc converter is the regulated dc voltage acting as the source for the voltage source inverter. The amplitude modulation of dc to ac conversion is fixed, therefore the dc link voltage should be almost constant if the control strategy is to produce the required inverter outputs.

A three-phase diode rectifier (International Rectifier 36MT80 35A) is selected for ac-dc conversion at the generator terminal. The semiconductors used for the dc-dc converter are a diode (RHRG30120) and an IGBT (IXSH40N60). A three phase IGBT inverter module (Semikron SKM40GD123D) is used as the dc-ac converter to supply the ac output voltage to the load. The module is made up of six IGBT switches with in-built free wheeling diodes and they are arranged in a three-phase bridge configuration. Each IGBT has a maximum collector-emitter voltage rating of 1200 V and a continuous collector current rating of 40 Amperes (at case temperature of 25°C).

The objectives of the experimental study include examining the validity of the chopper control scheme and the effectiveness of the CPLD chopper controller; the validity of the variable speed control scheme, the performance of the FPGA variable speed controller and a performance study of the overall control system. The two control

systems, chopper controller and variable speed controller, are discussed in detail in separate sections with the associated measurement and driving circuits.

The overall test system

An integrated physical model including the FH2 test bed, the power electronic system, and the developed control system is constructed to investigate and assess the operation and performance of the developed control strategy. The schematic diagram of the overall test system is shown in Figure 9.2. The overall integrated physical model includes the CPLD/FPGA implemented controllers.

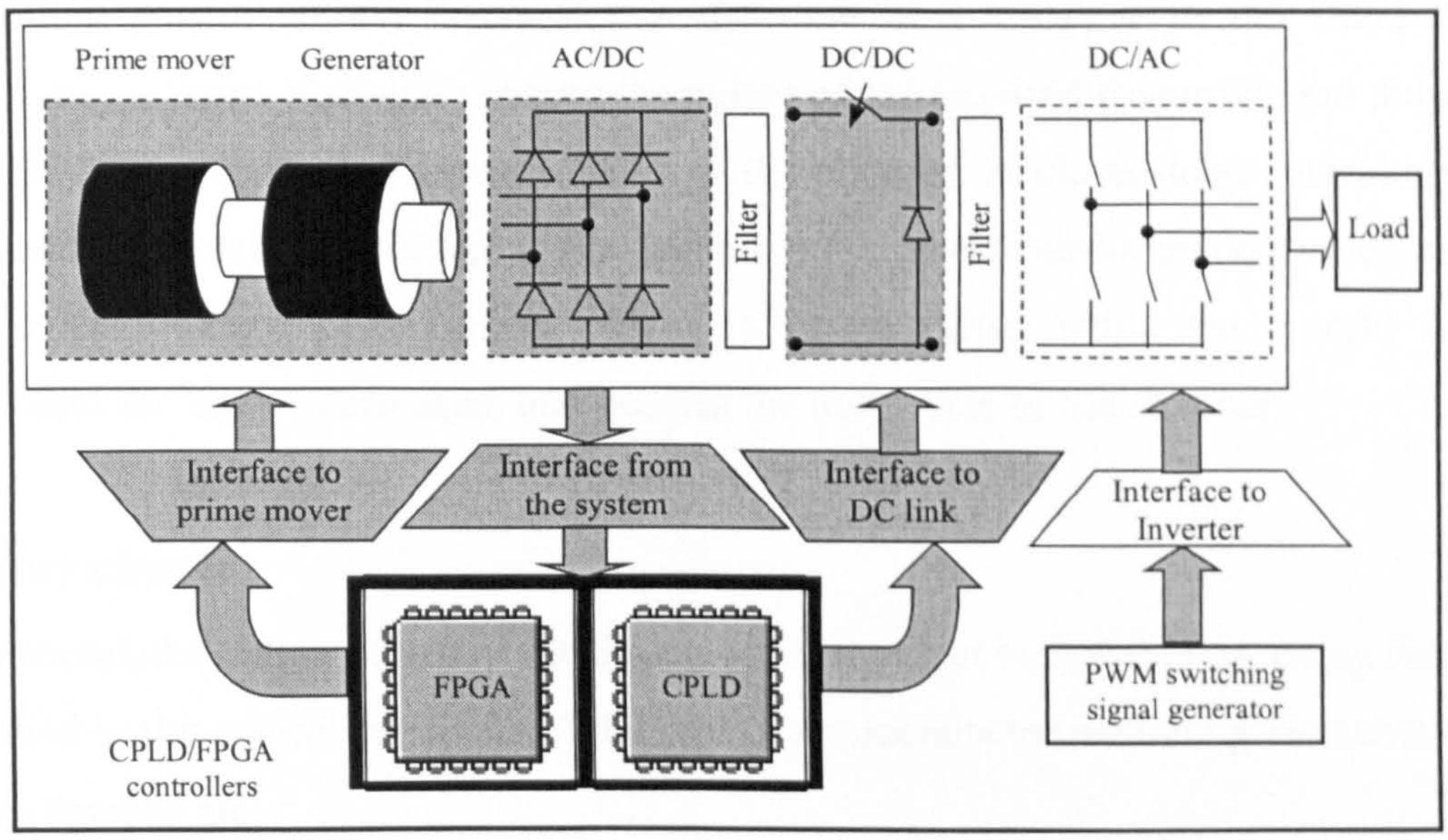


Figure 9.2. Overall integrated physical model with CPLD/FPGA implemented controllers

9.2. Chopper control system and experimental study

9.2.1. Chopper control system circuit

A chopper converter, like other power electronic converters, has a power circuit section and a switching control circuit section. The power circuit section, which normally has to handle relatively large voltages and currents, has been described in earlier chapters. The function of the switching control circuit section in the dc-dc

converter is to provide the appropriate control signal to the control terminal of the power semiconductor in the converter so that the power semiconductor can be turned on or off with the correct timing to enable the converter to operate in the required manner. The switching control section, generally consisting of integrated circuits and other low power electronic components, normally works from low voltage power supplies.

Three functional circuit blocks within the switching control section are shown in Figure 9.3. These are labelled as the control block, the timing block and the driver block. The timing block generates the timing pulses, according to the required switching pattern of the converter. In the case of a chopper in the stand-alone generating system, it should produce the pulses of the required frequency and duration needed for the controlled semiconductor in the chopper. A closed-loop control circuit is used to maintain the voltage at the desired value. The closed-loop controller senses the actual voltage, compares it with a reference value, while the control block processes the input information and decides the duty cycle of the chopper.

Driver circuit

In general, the output pulses of the timing block may not be suitable for being directly applied to the control terminals of the power semiconductor switching elements. The main reasons are:

- The power capability of the timing pulses may be insufficient. The output of the control signal is at TTL level. This is not enough to drive the IGBT, therefore, the first function of the driver circuit is to step up the driving ability. The maximum threshold voltage of the IGBT is 6.0 volts. In this design, the driver circuit produces switching signals at 15V, which is significantly higher than the threshold voltage of the device.
- There is usually a need to provide electrical isolation between the logic circuit and the power semiconductor switching element.

The driver circuit block serves to meet the above requirements. Figure 9.4 shows the circuit diagram of the driver circuit. The transformer provides isolation to the circuit

so that the reference level (ground) for the outputs can be different. The diode bridge is used to obtain a DC supply for the driving circuit. The voltage regulator maintains a steady 15V DC supply to the circuit. The optical isolator provides complete isolation of the input signals from driver circuit. The IGBT gate driver MC33153 supplies the driving voltage and current to the IGBT.

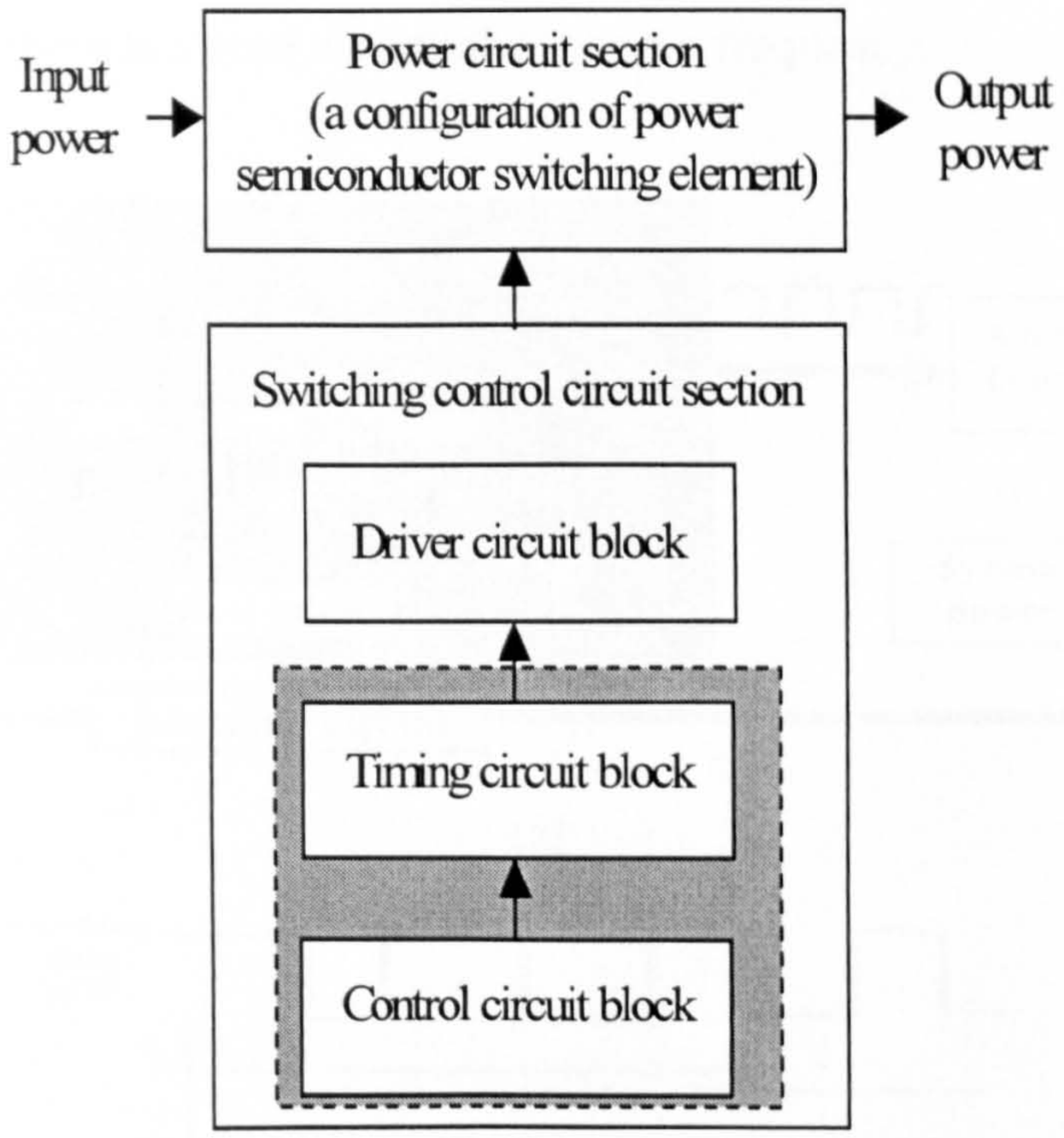


Figure 9.3. Block diagram of the switching control section of a chopper circuits

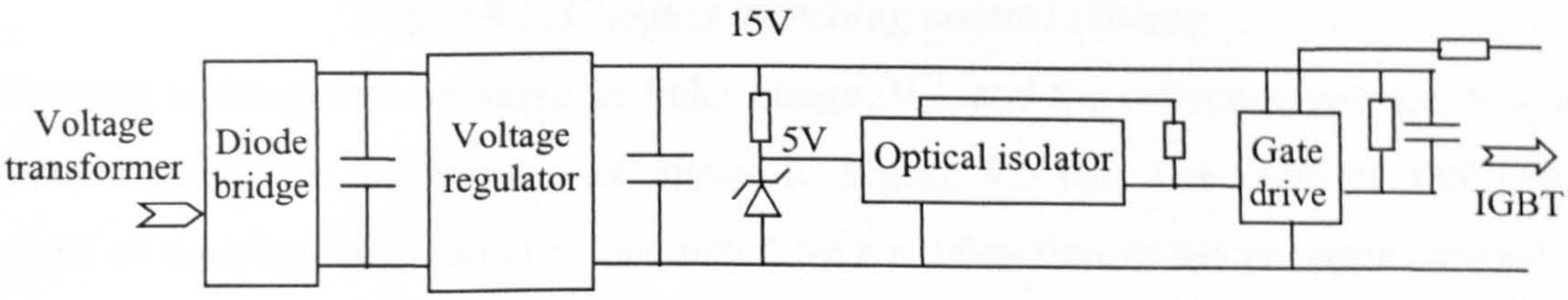


Figure 9.4. IGBT driving circuit

Switching control

The chopper circuit is switched at a constant frequency (hence, a constant switching time period $T_s = t_{on} + t_{off}$). The ON duration of the switch can be adjusted to control the average output voltage. The switch duty ratio D, which is defined as the ratio of

the ON duration to the switching time period, is variable, so that the average output voltage is controlled.

The schematics of the control scheme for generating the timing pulses is shown in Figure 9.5 (a). The timing requirements of CPLD, A/D converter and the IGBT, the switching frequency of the chopper can be set by programming the CPLD and easily reprogrammed if there is a need to vary the chopper frequency.

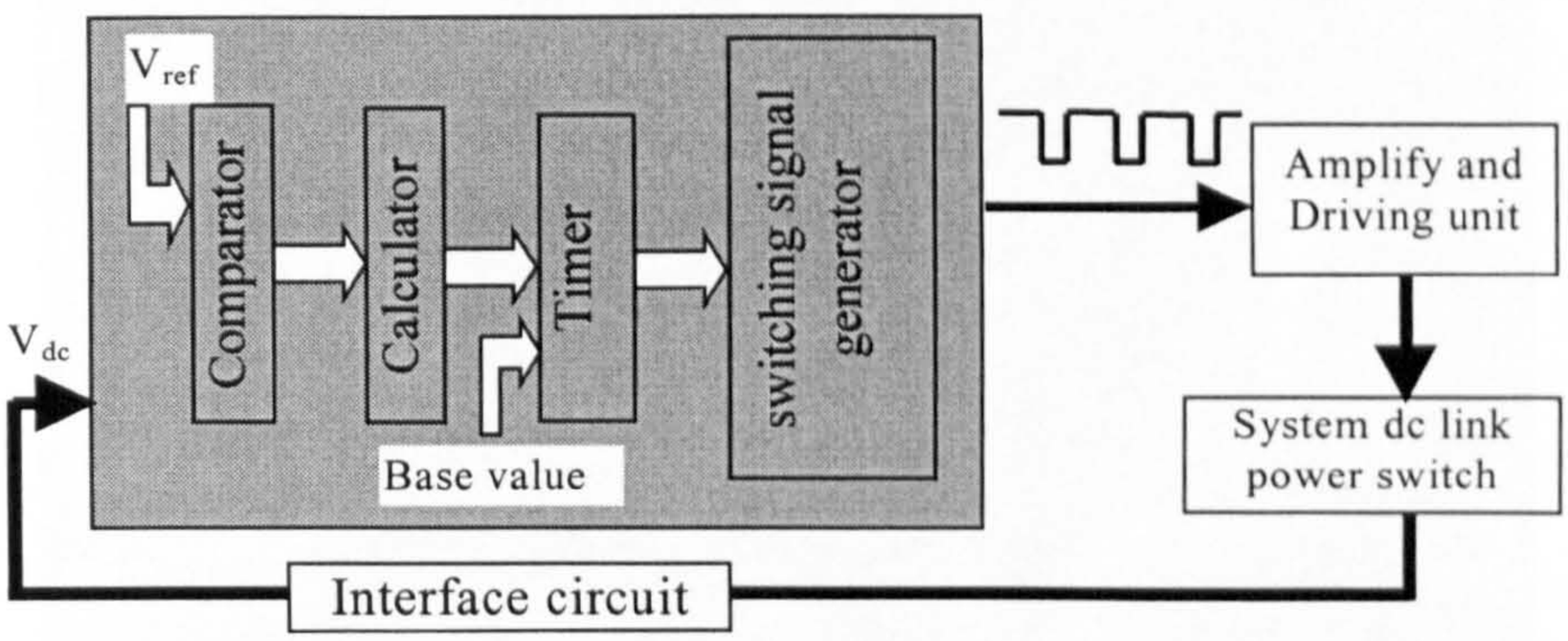


Figure 9.5. Chopper switching control scheme

The error between the measured dc link voltage, $V_{dc,}$ and the reference voltage, $V_{ref,}$ is obtained using the comparator as shown in Figure 9.5 (a). The required ON state length of the switching signal is calculated by a subfunction in the chopper controller and is consequently used to set the timer on the basis of a base value. In this way the duty cycle ratio, D, of the chopper is varied. Then the switching signal is generated by the signal generator. The ON period of the IGBT in the chopper circuit is the width of the timing pulses from the switching signal generator. The operation of the circuit is further made evident by the relevant waveforms in Figure 9.5 (b).

CPLD system

In the circuit design, both timing circuit and control circuit are implemented in a CPLD XC95108-PC84 chip, a member of the Xilinx XC9500 family of CPLD. The implemented CPLD chip is mounted on a Xilinx XS95 development board as shown in Figure 9.6.

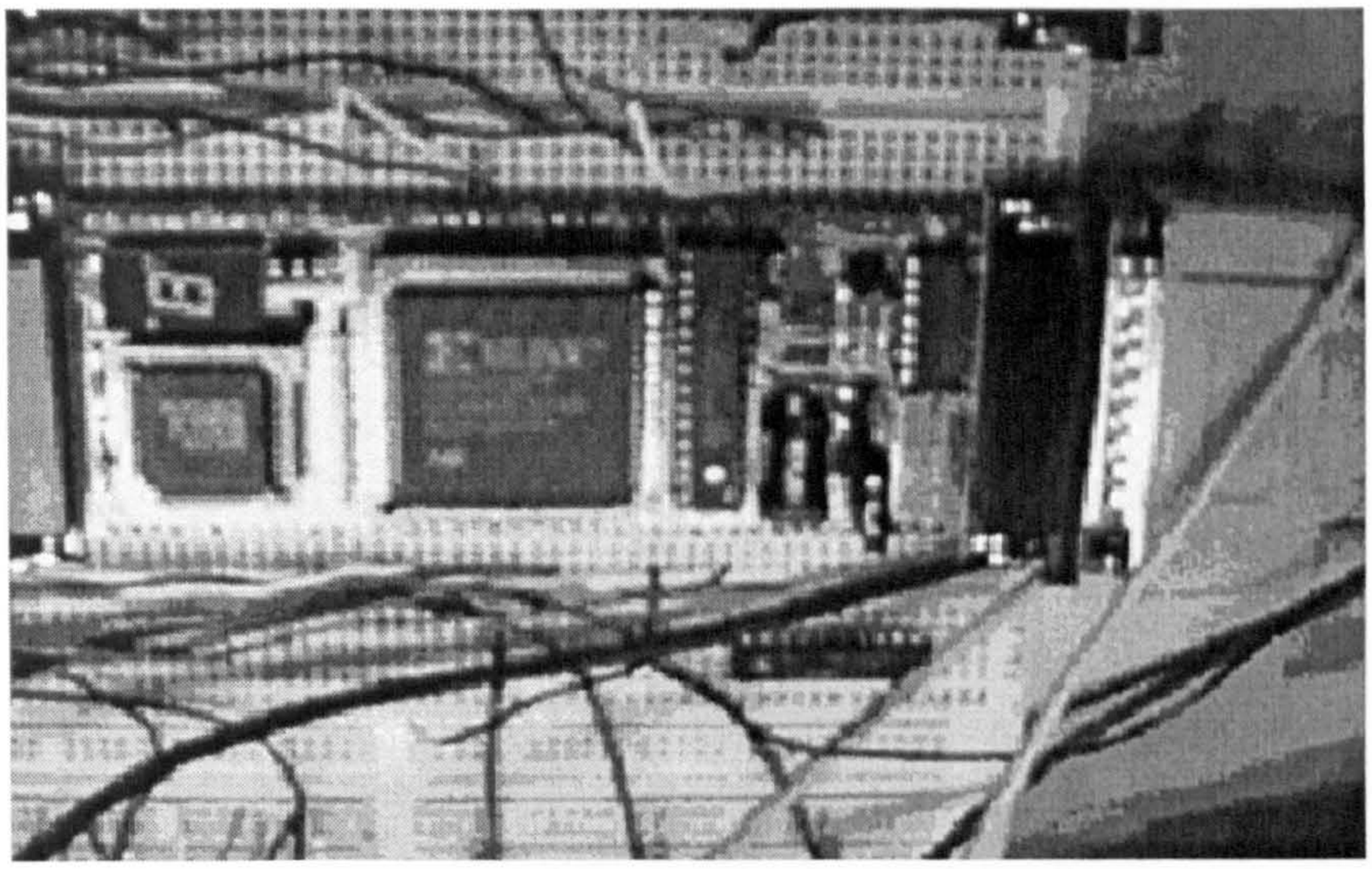


Figure 9.6. Xilinx XS95 development board

Signal measurement and conversion

As the CPLD XS95 development board is 5V regulator configured, the inputs of the control signal are at digital signal level. The dc link voltage has to be scaled down into the range of digital signal level and converted to a digital signal. Figure 9.7 shows a voltage transducer for measurement and an A/D converter used for converting the signal to 8-bit digital.

The voltage sensor used in this project is a PCB-mounting Hall-effect voltage transducer (LV25-P), which provides a measured voltage signal to the A/D converter, the LV25-P also provides the necessary galvanic isolation between the power circuit and the control system.

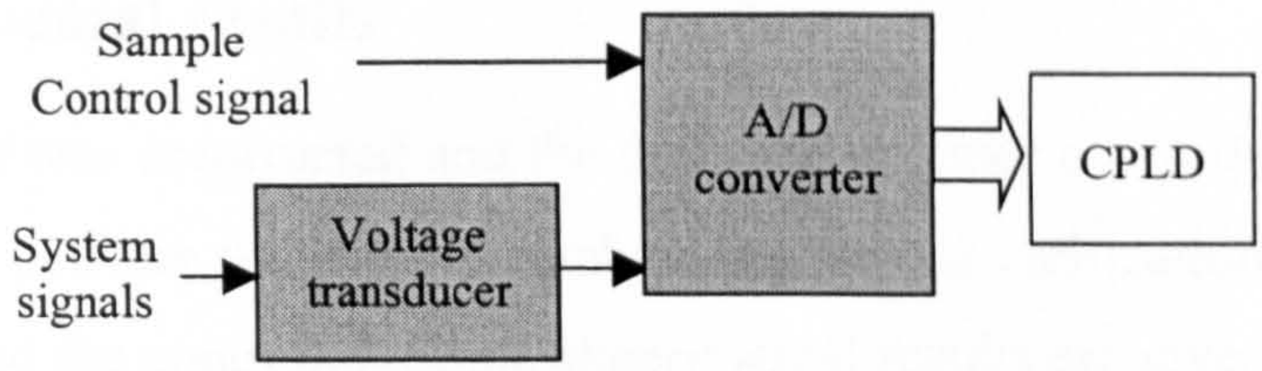


Figure 9.7. Signal measurement and conversion system

An A/D converter (National Semiconductor ADC0804) is used for the conversion of the analogue signal from the transducer into the digital signal. The device is an 8-bit CMOS A/D converter based on the successive approximation conversion technique. The logic inputs meet TTL specifications as well as CMOS. The READ signal initiates the A/D conversion process in the ADC0804 while the WRITE signal enables the CPLD to sample the converted digital signal from the A/D converter. The sequence of these two signals is set so that the CPLD takes a sample of the digital value after each time of conversion. The A/D conversion control signals are produced by the same CPLD chip so that it can be easily adjusted to suit to the A/D conversion speed. The advantage of this timing control method is that there is no need for a separate clocking circuit, hence the complexity of the circuit is reduced. Figure 9.8 shows a simplified schematic diagram of the signal sensor and conversion circuit. The buffer 74HC541 is used to increase the driving ability of the ADC0804 output signals.

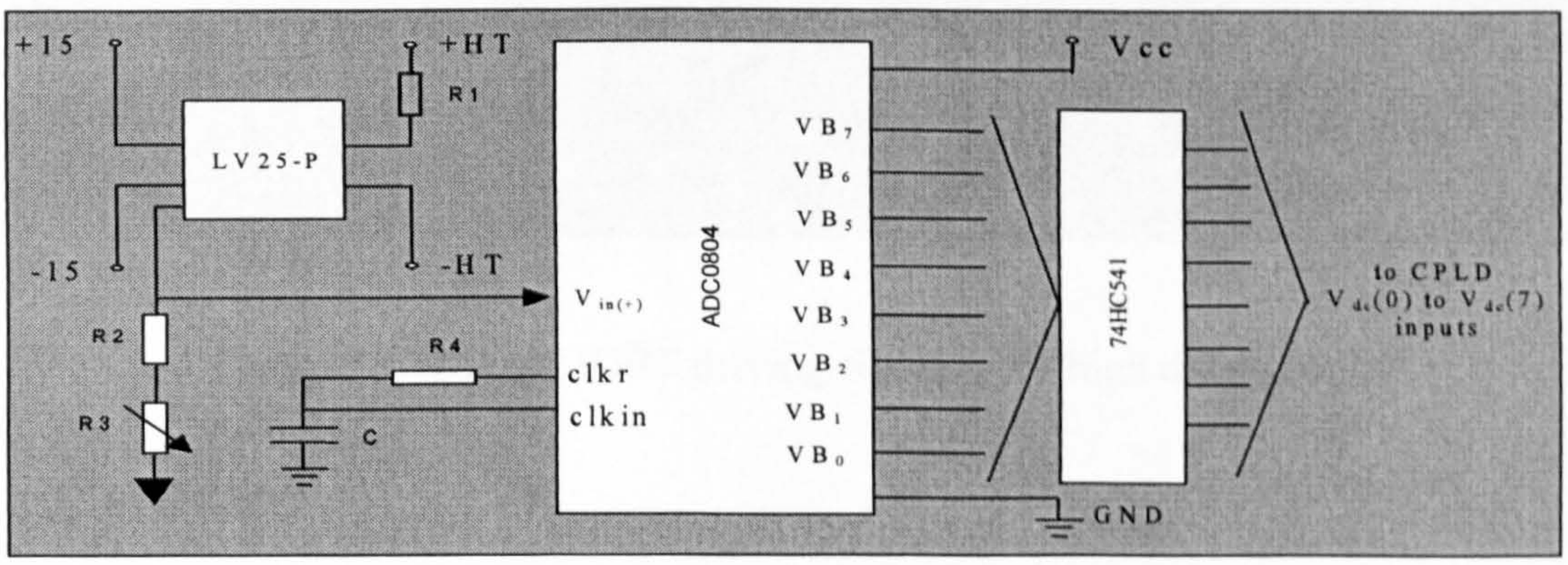


Figure 9.8. Interface circuit schematic

9.2.2. Experimental results

A chopper circuit was constructed and the designed chopper controller was connected to the chopper to perform the experimental studies for the verification of the functions of the chopper and the controller. Some experimental results are given in Figure 9.9 to Figure 9.12, where the waveform sequence (top to bottom) is driving signal, inductor current, IGBT current and diode current. The upper and lower limits of the duty ratio are set as 87.7% and 6.7% respectively. The results show that chopper system works well over the range of duty ratio.

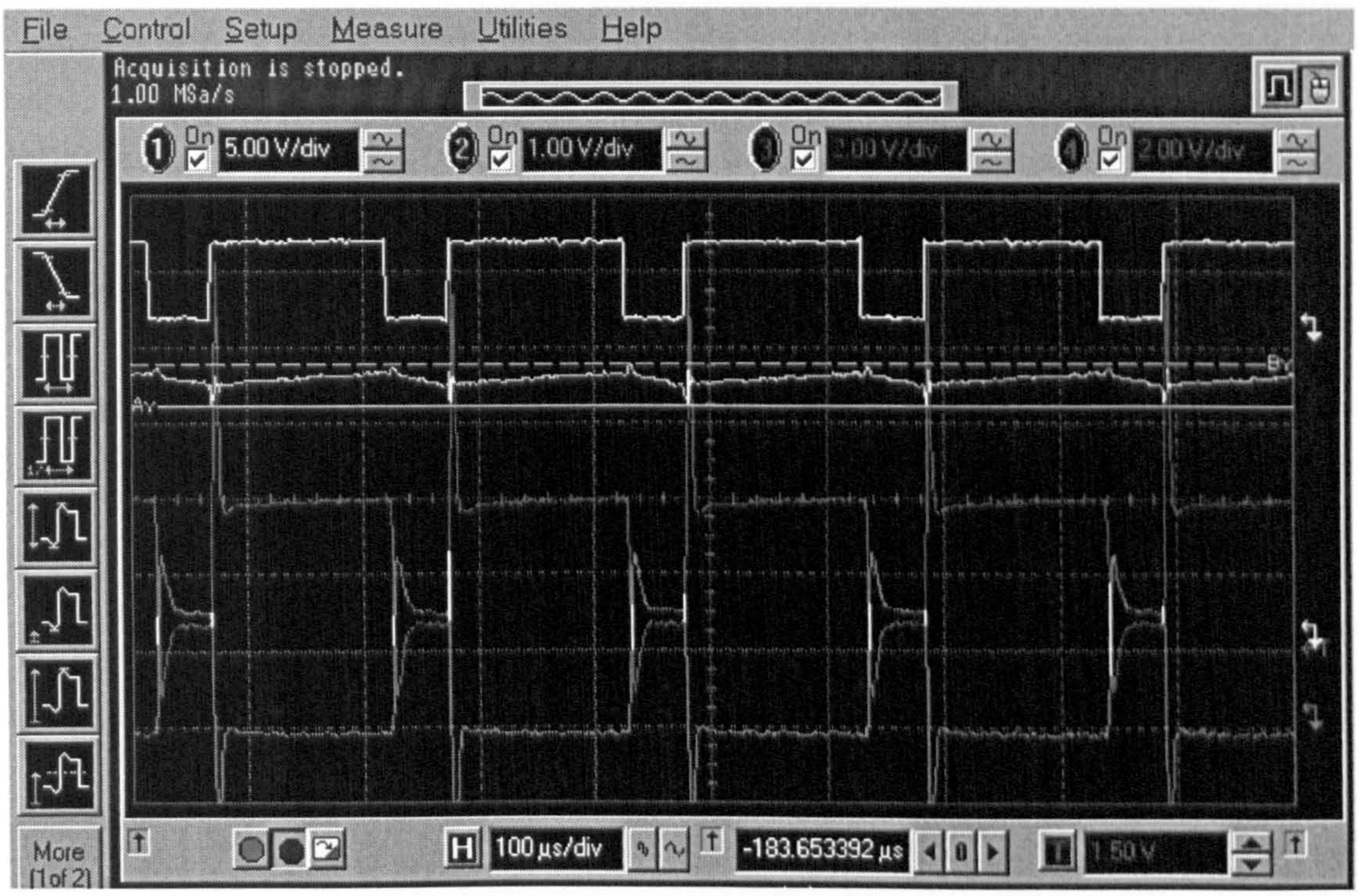


Figure 9.9. Voltage IGBT driving signal (very high duty ratio D)

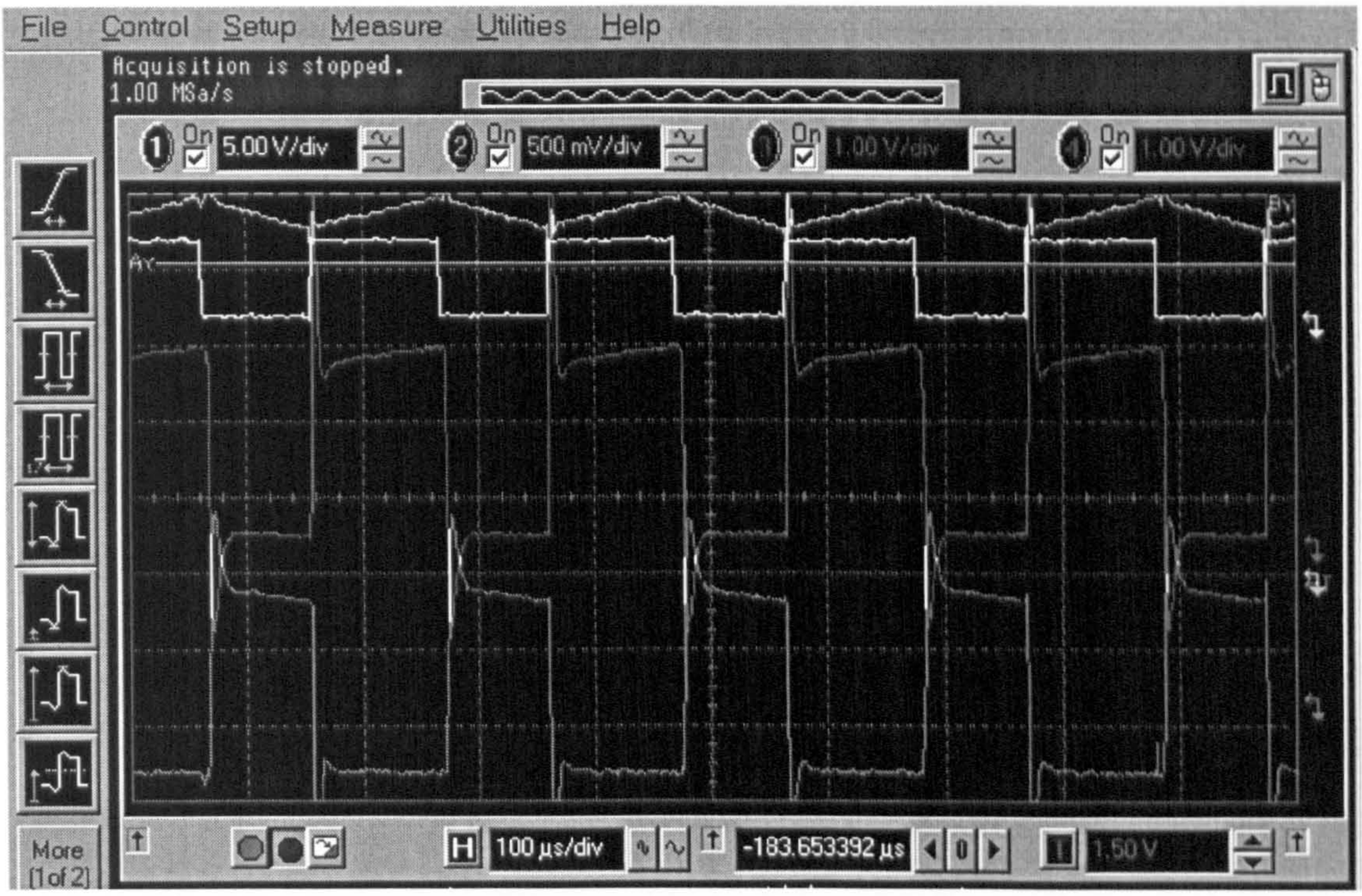


Figure 9.10. Voltage IGBT driving signal (high duty ratio D)

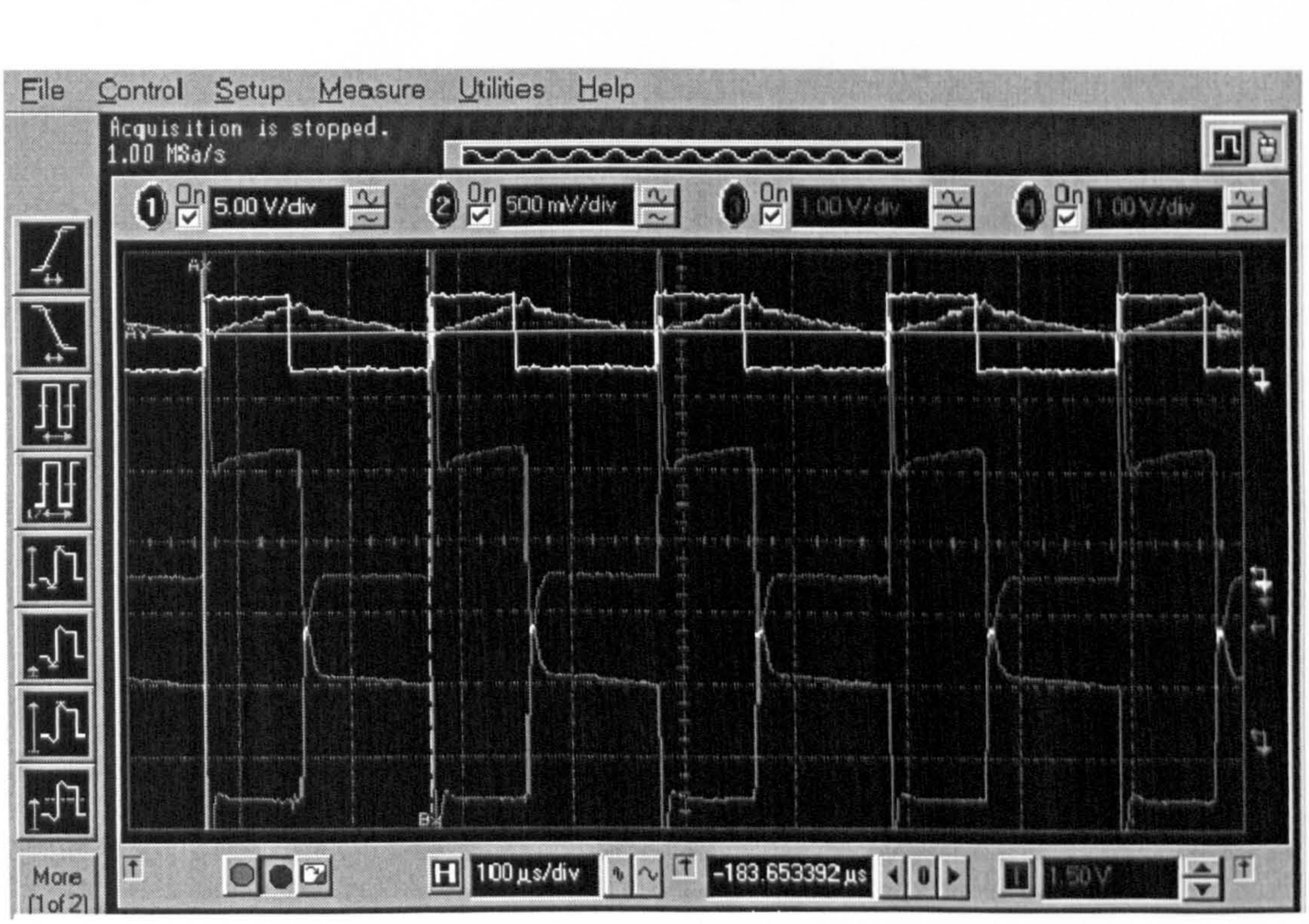


Figure 9.11. Voltage IGBT driving signal (low duty ratio D)

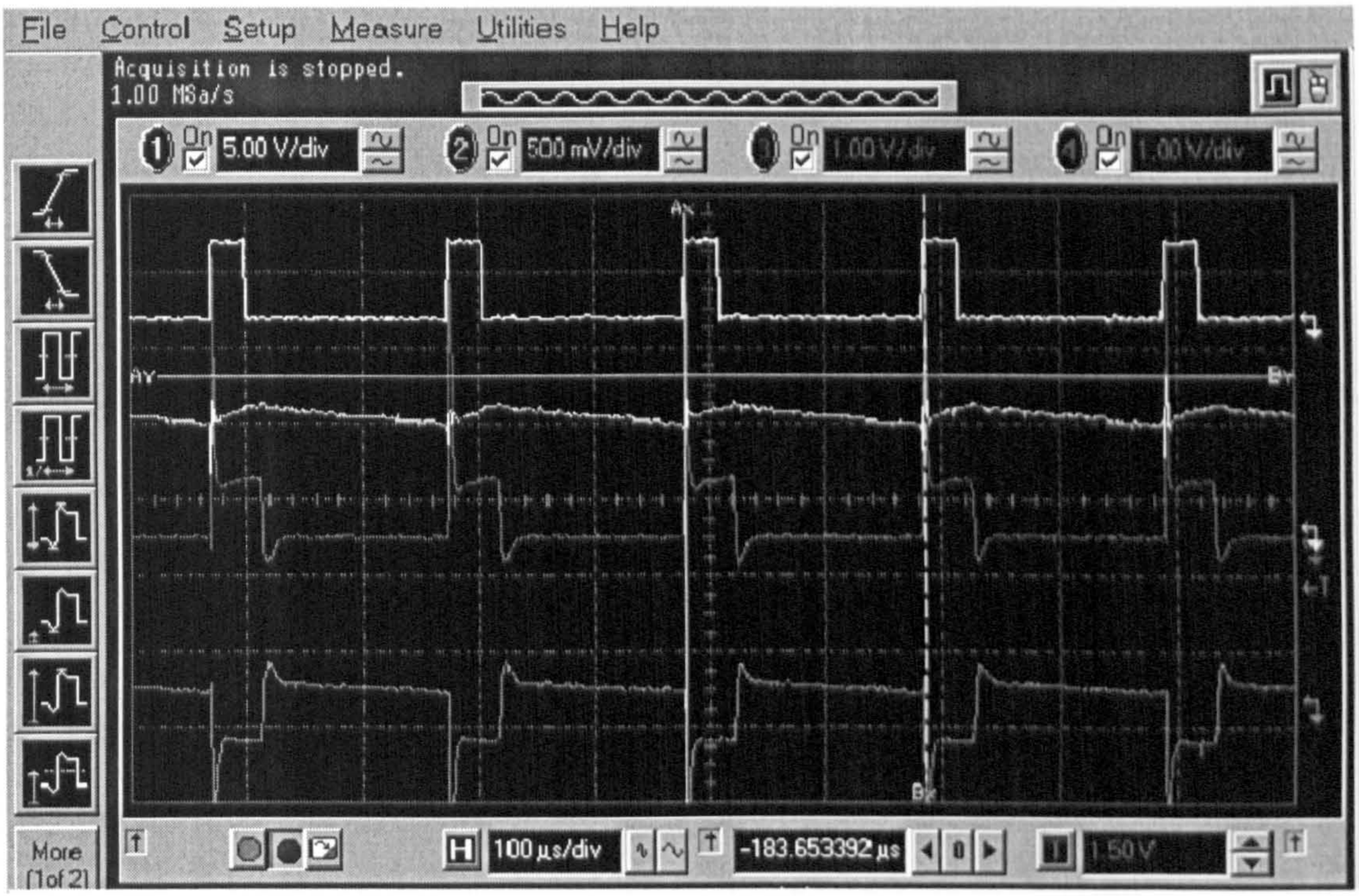


Figure 9.12. Higher voltage IGBT driving signal (very low duty ratio D)

9.3. FPGA variable speed control system

The behaviour of the FPGA variable speed controller is tested in the modelled stand alone power generation system. An increase of fuel input is simulated with a decrease of the braking torque applied to the machine. The test system is used to validate the developed control system based on FPGA.

9.3.1. FPGA variable speed control system circuit

The FPGA variable speed control system circuit includes a programmable gate array chip XC4010PC84, the interface circuits, amplifying circuit, and filtering circuits. The control system block diagram is shown in Figure 9.13.

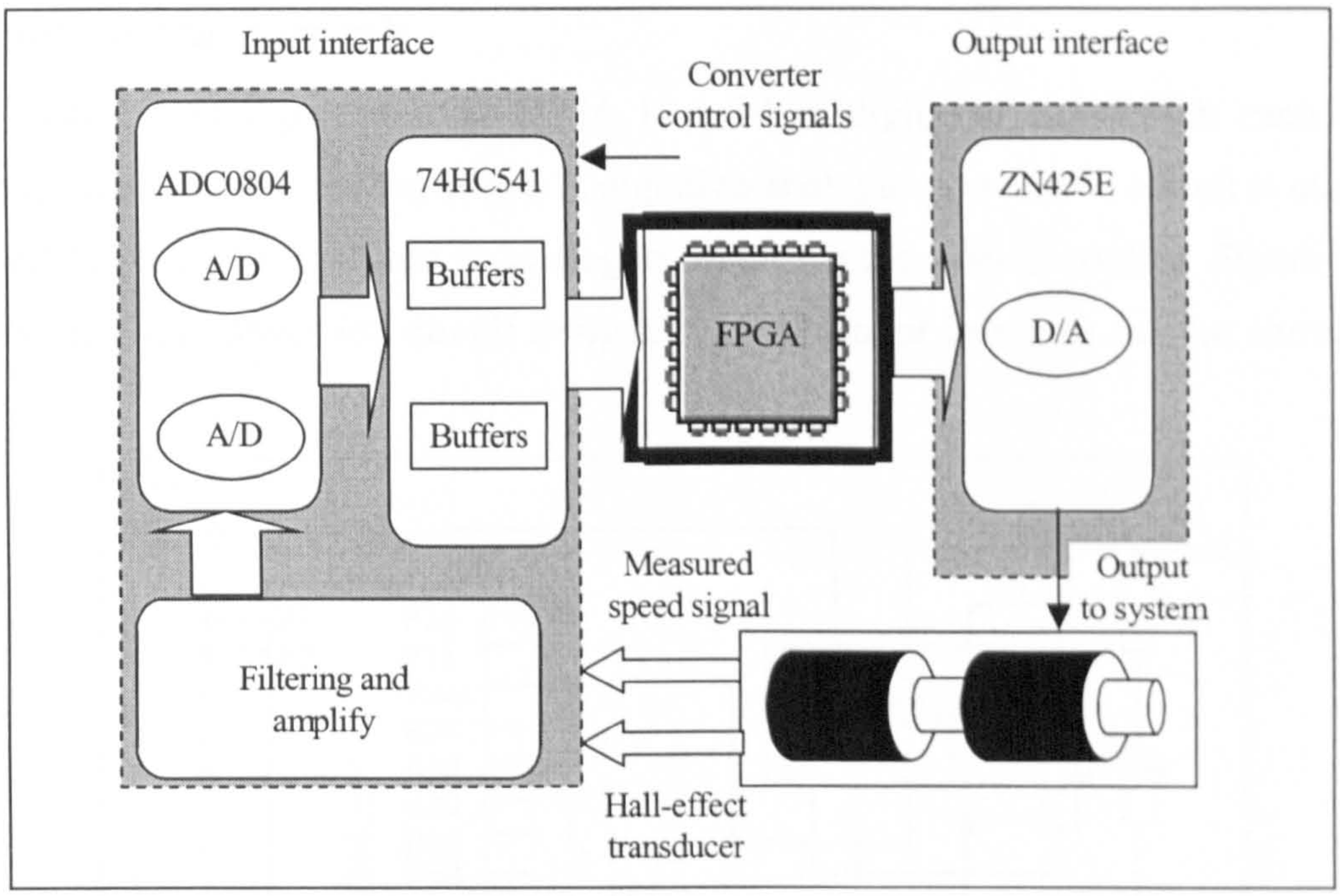


Figure 9.13. Variable speed control system block diagram

Inputs interface circuit

The speed controller has two input signals, current and machine speed. The inputs interface circuit includes two A/D conversion sub-circuits, the current signal conversion and the speed signal conversion. The two sub-circuits are identical. Figure 9.14 shows the connection of the A/D conversion circuits.

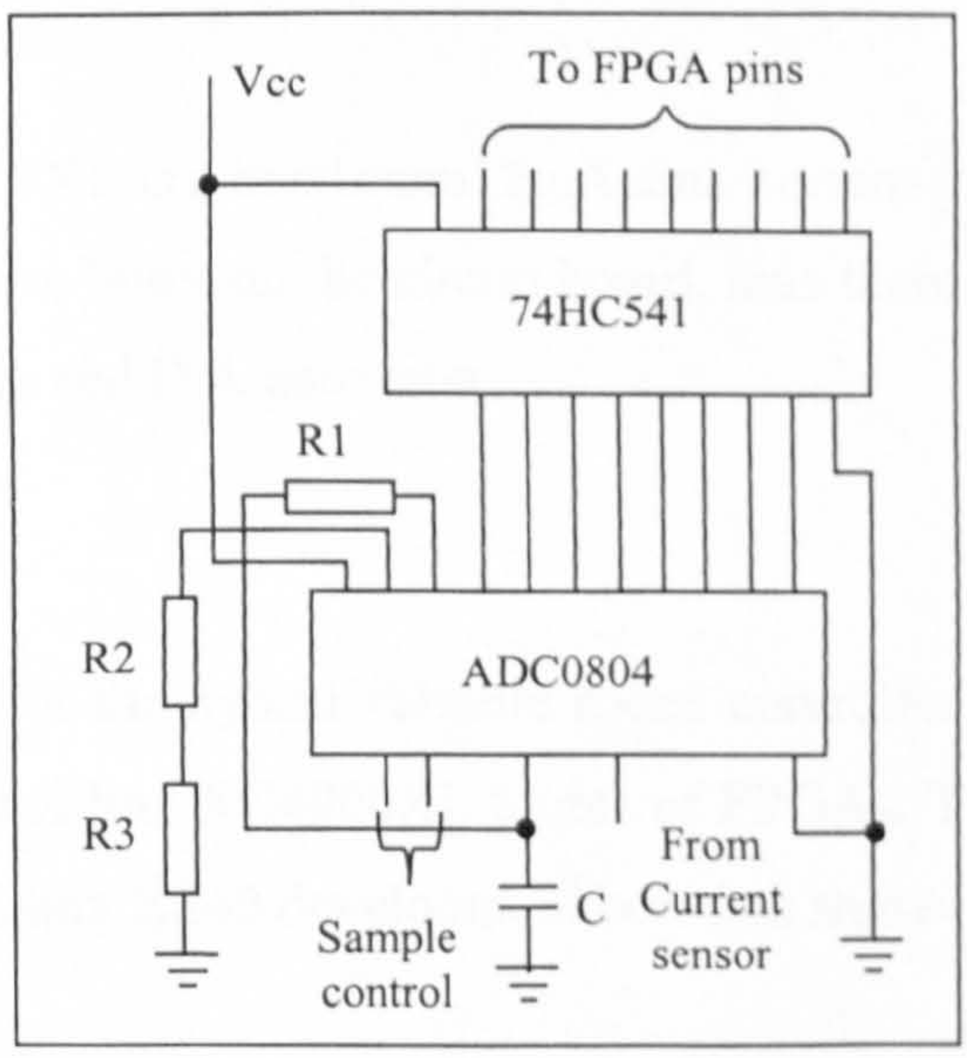


Figure 9.14. A/D conversion circuit

Output interface circuit

The control signal generated in FPGA is a 8-bits digital signal, which cannot be directly applied to the control target. A digital to analogue conversion circuit is used to convert the digital signal into an analogue signal for the system control. Figure 9.15 shows the D/A conversion circuit designed as the output interface for the controller test.

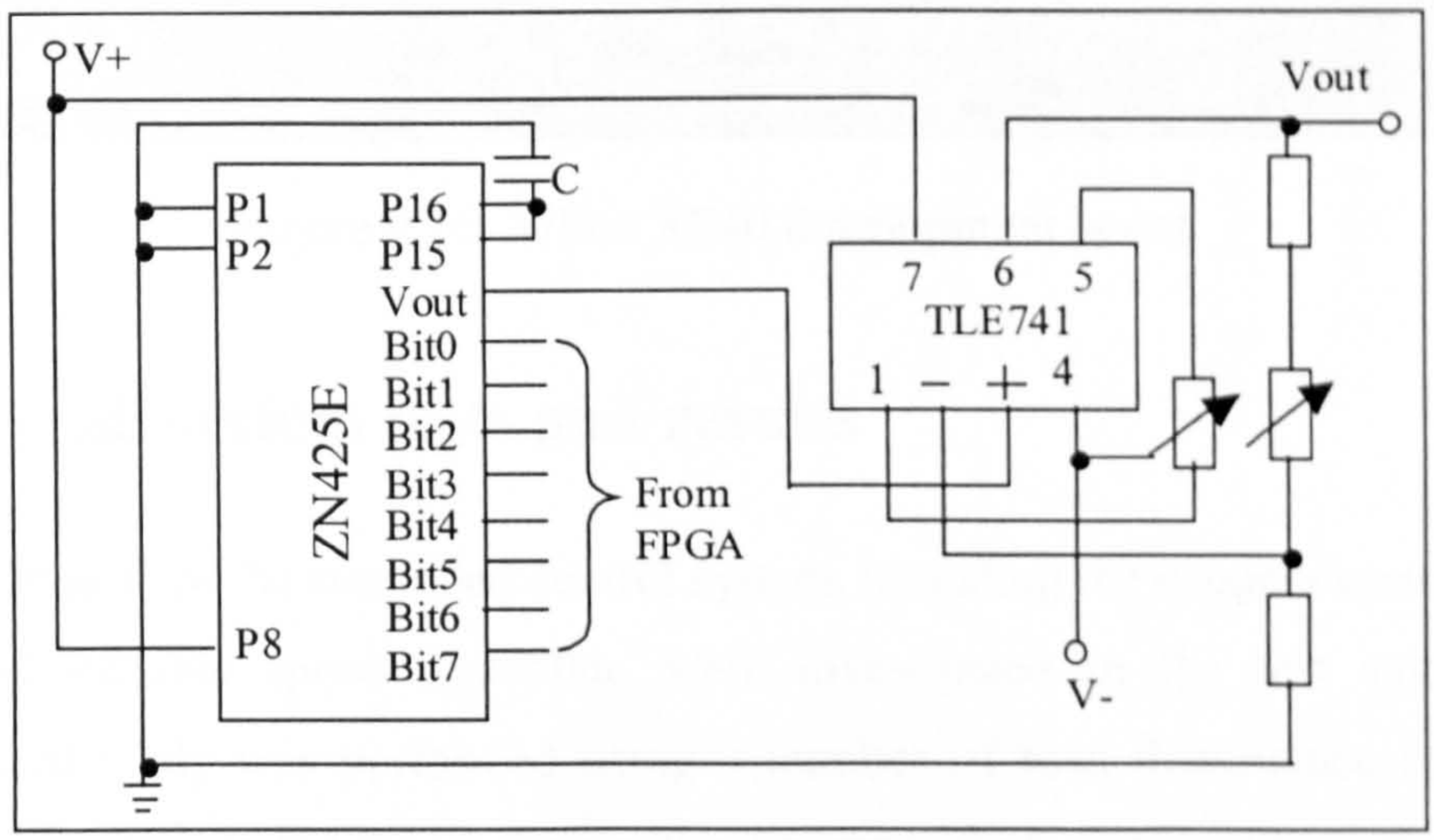


Figure 9. 15. The D/A conversion circuit (output interface)

In Figure 9.15 the ZN425E converts the 8-bit digital signal and produces an analogue voltage output. The buffer amplifier is connected for removing the offset voltage and calibrating the converter.

A feature of the XC4000XL is a maximum12mA sink current per output and of course the chip can directly drive buses on the circuit board, thus there are no external buffers needed between the chip and D/A converter.

FPGA circuit

The target technology for the hybrid variable speed controller design is XC4010XL-PC84, a member of the Xilinx XC4000XL Series of FPGAs. The implemented FPGA chip is mounted on a Xilinx XS40 development board as shown in Figure 9.16.

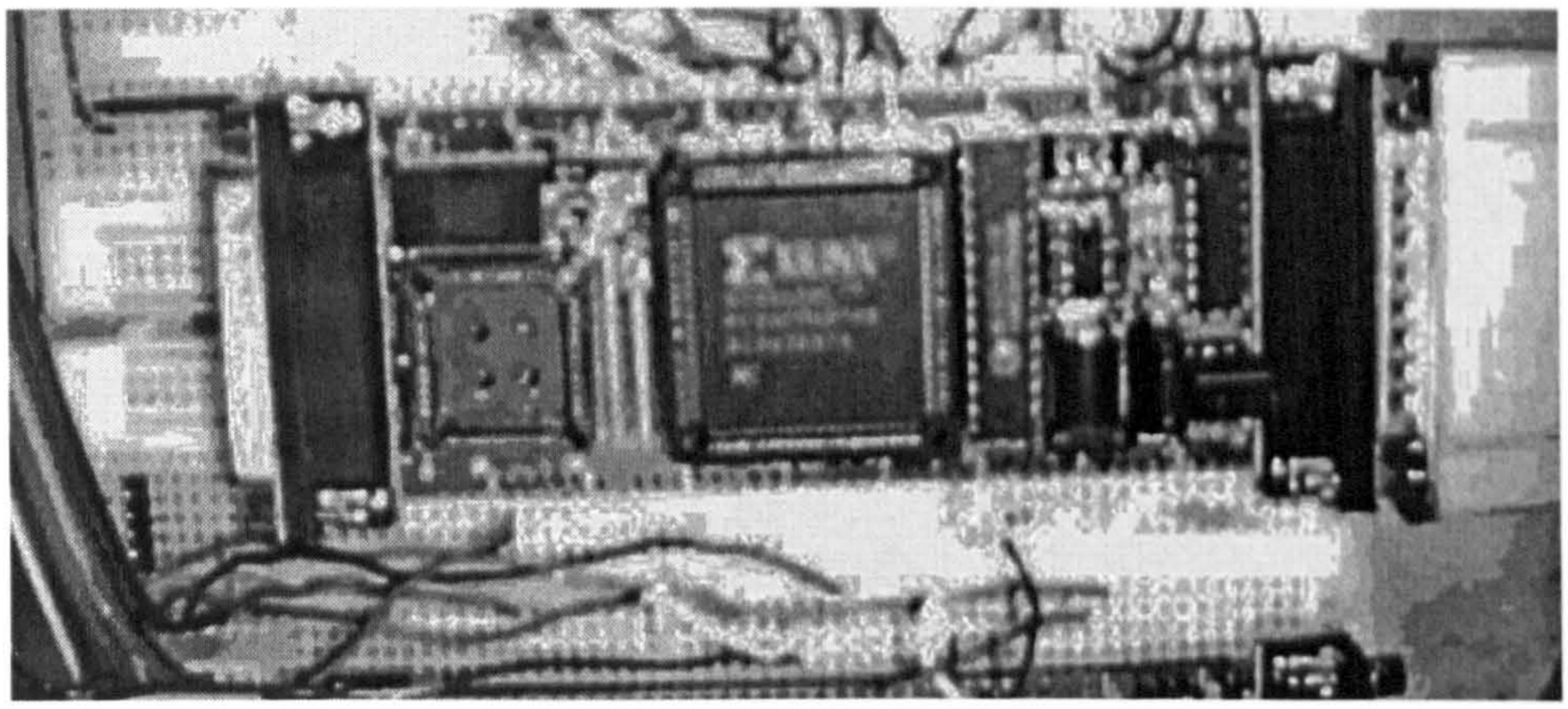


Figure 9.16. Xilinx XS40 development board

9.4. Overall system tests and results

The performance of the combined control system, including the chopper controller and the hybrid variable speed controller, were investigated in the test system. The experimental study was performed using a number of load disturbance conditions. Disturbance was created by changing the resistive load connected at the chopper output. Figures 9.17 and 9.18 present the system responses without the presence of the proposed control system, while the system responses with the proposed control system are shown in Figures 9.19 and 9.20.

System performance without the proposed control systems. The no load operating speed of the generator was set as 1500rpm. The operation of the speed controller is restricted by setting zero braking torque. To restrict the operation of the chopper control system, the switching duty ratio of the chopper controller is fixed at 50%. The load current varies between two different loading levels as shown in Figures 9.17 (a) (55% rising to 100%) and Figures 9.18 (a) (100% falling to 59%). The corresponding machine speeds and load voltages are shown in Figures 9.17 (b), (c) and in Figures 9.18 (b), (c). It can be seen that the machine speed drops (100% down to 91%) and the output voltage decreases (100% down to 87%) in Figures 9.17 (b) and (c) when the load current increases. Figures 9.18 (b) and (c) show that the machine speed rises from 90 % rising to 100%) and so does the output voltage (87% rising to 100%) when the load current decreases.

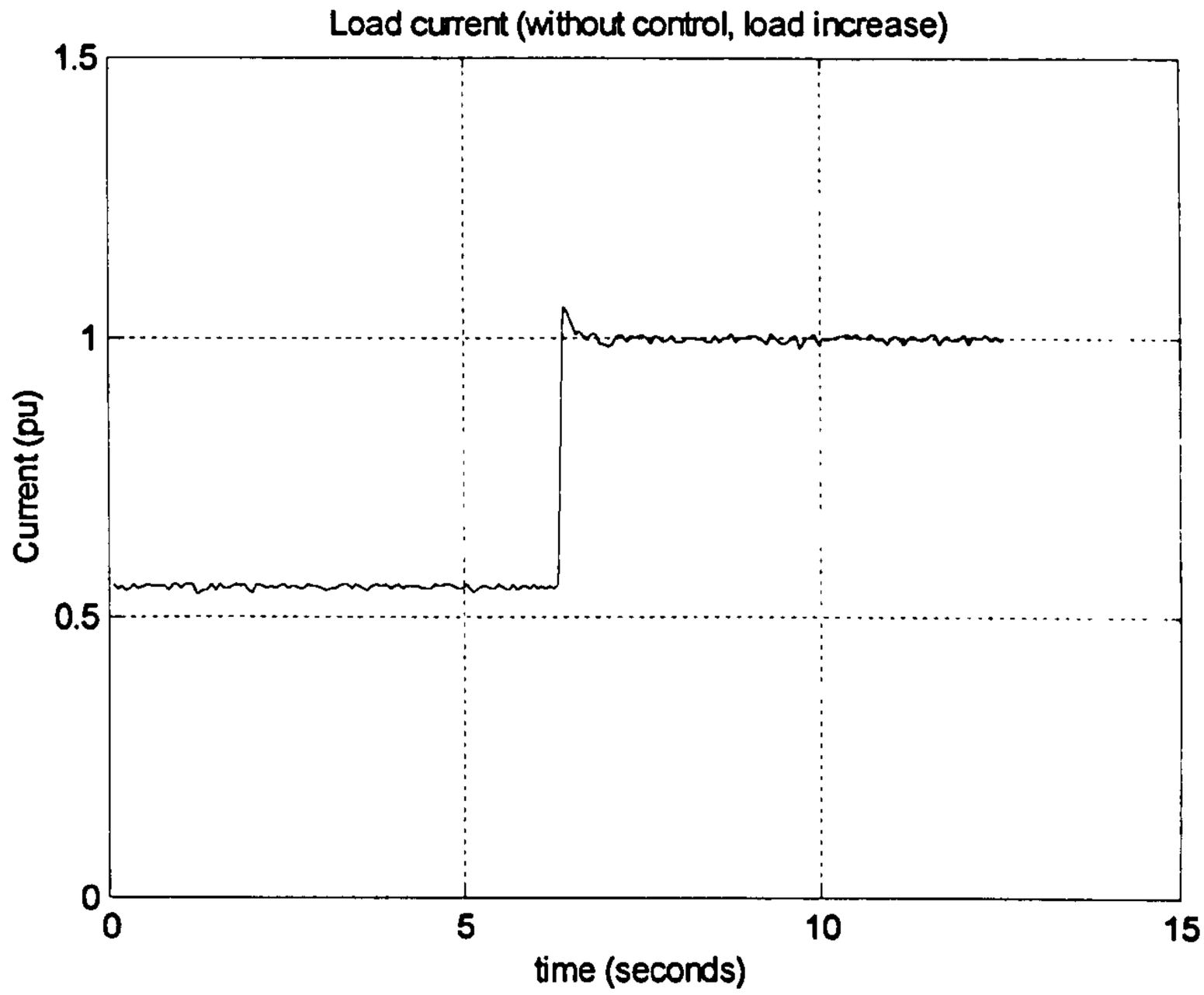


Figure 9.17. (a) Load current

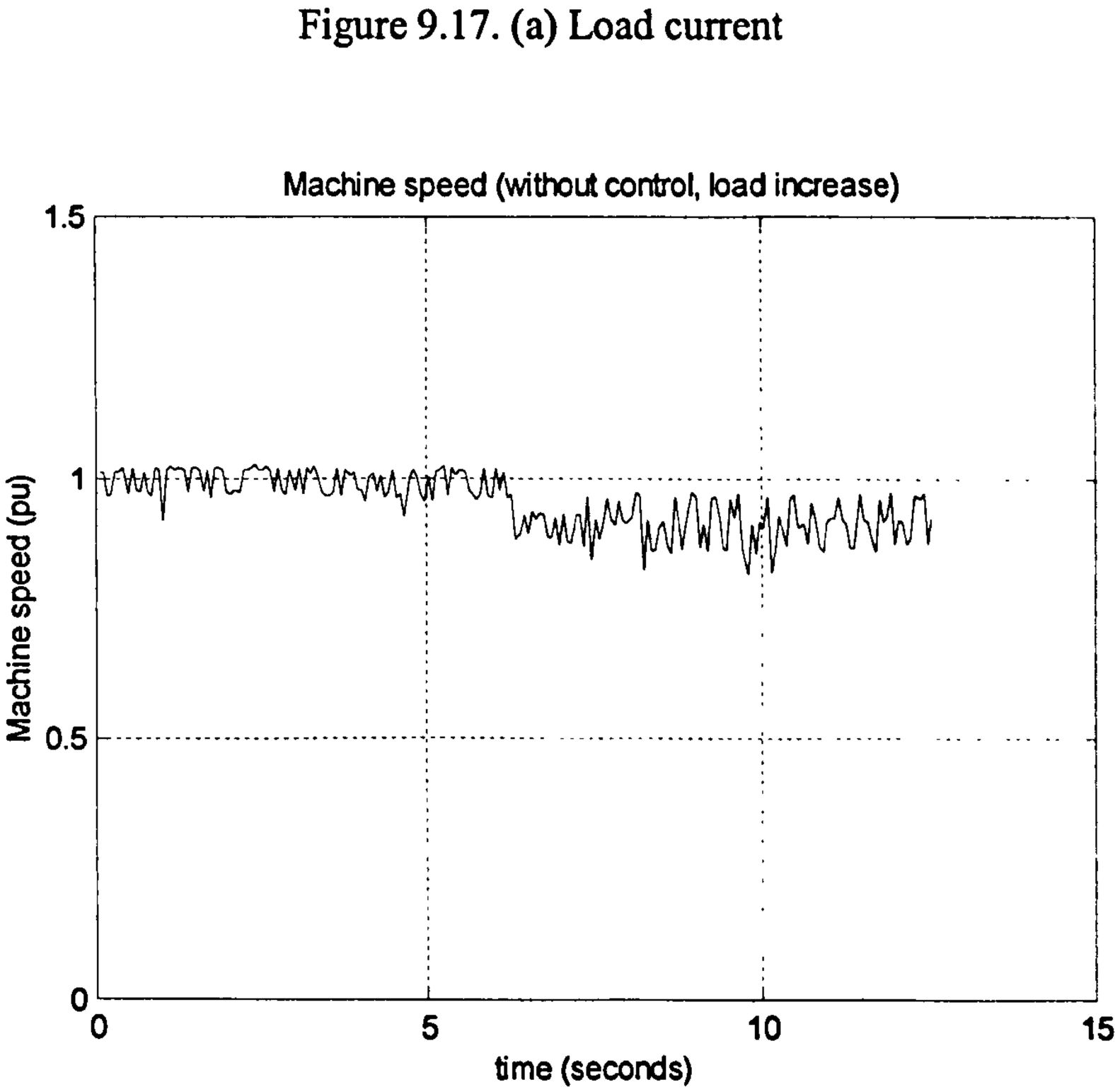


Figure 9.17. (b) Machine speed

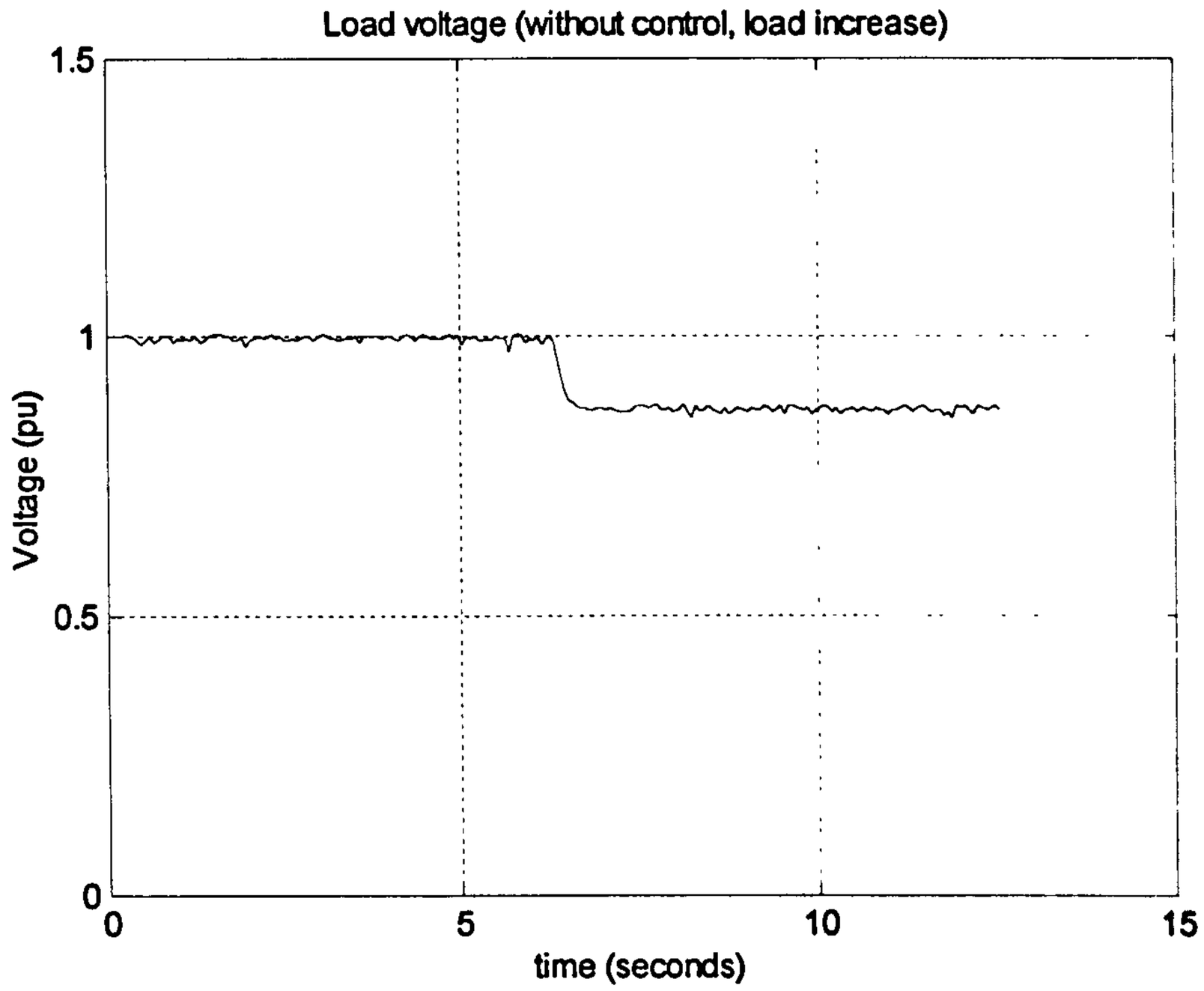


Figure 9.17. (c) DC/DC converter output voltage

Figure 9.17. Experimental results without control (load increase)

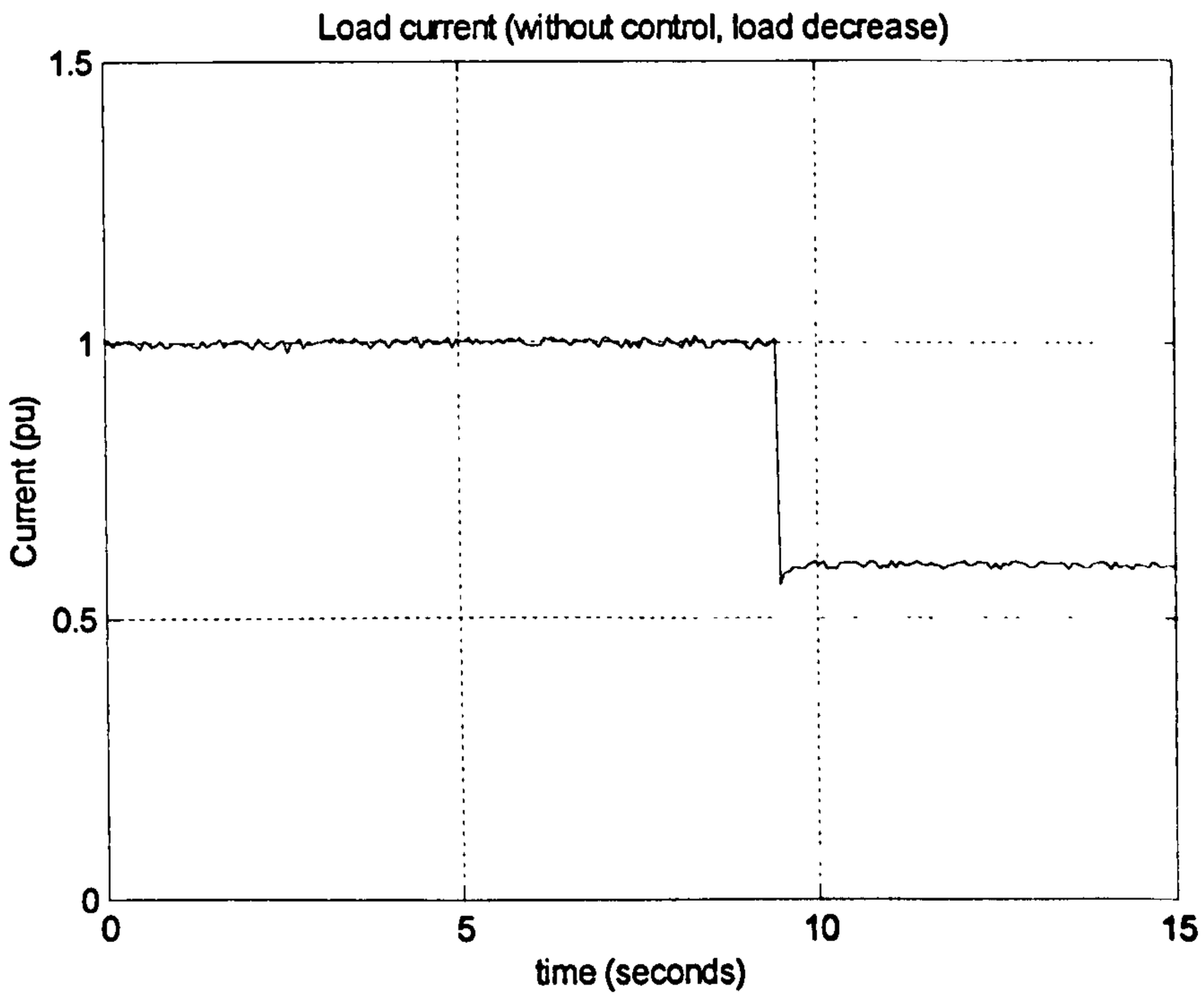


Figure 9.18. (a) Load current

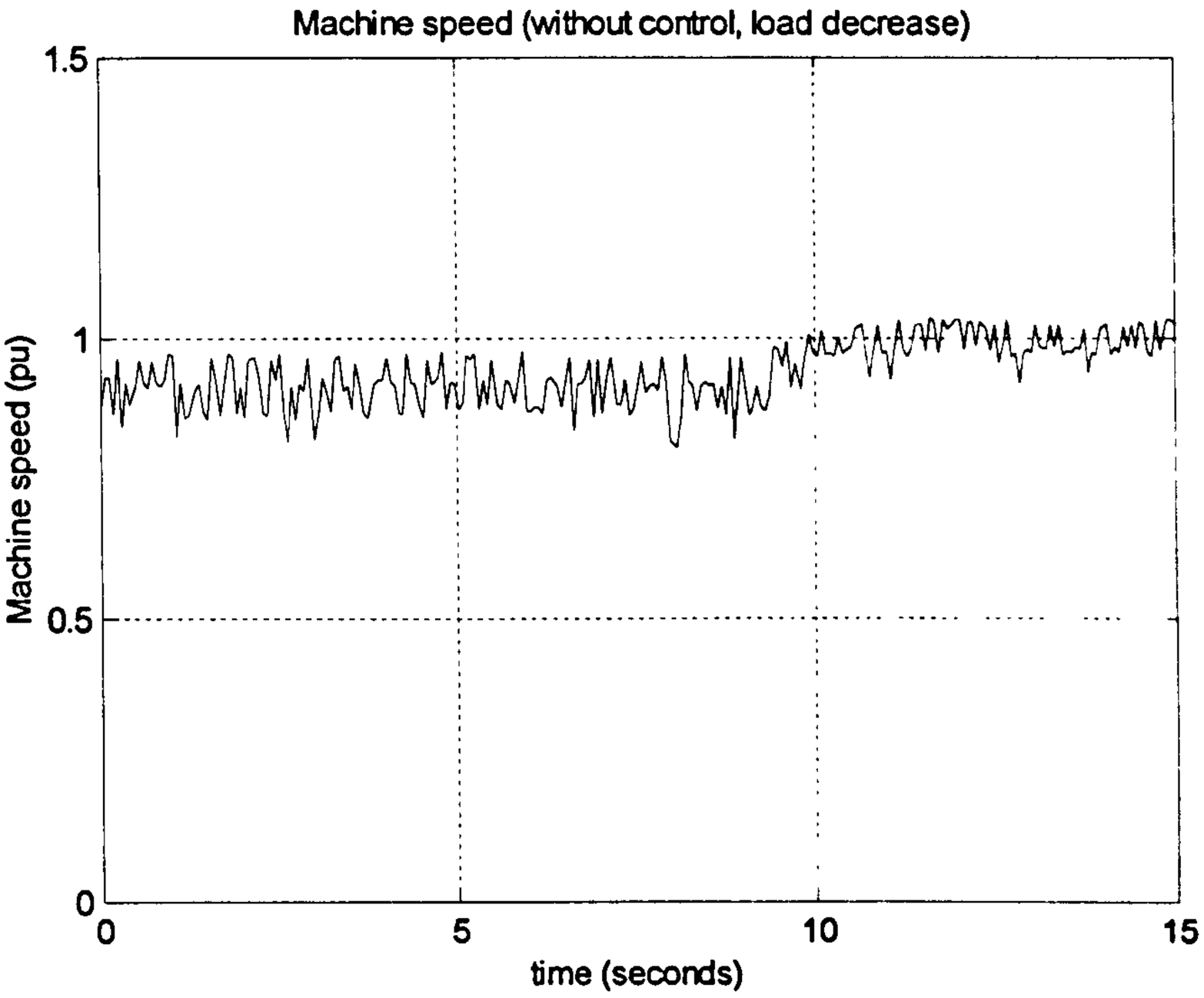


Figure 9.18. (b) Machine speed

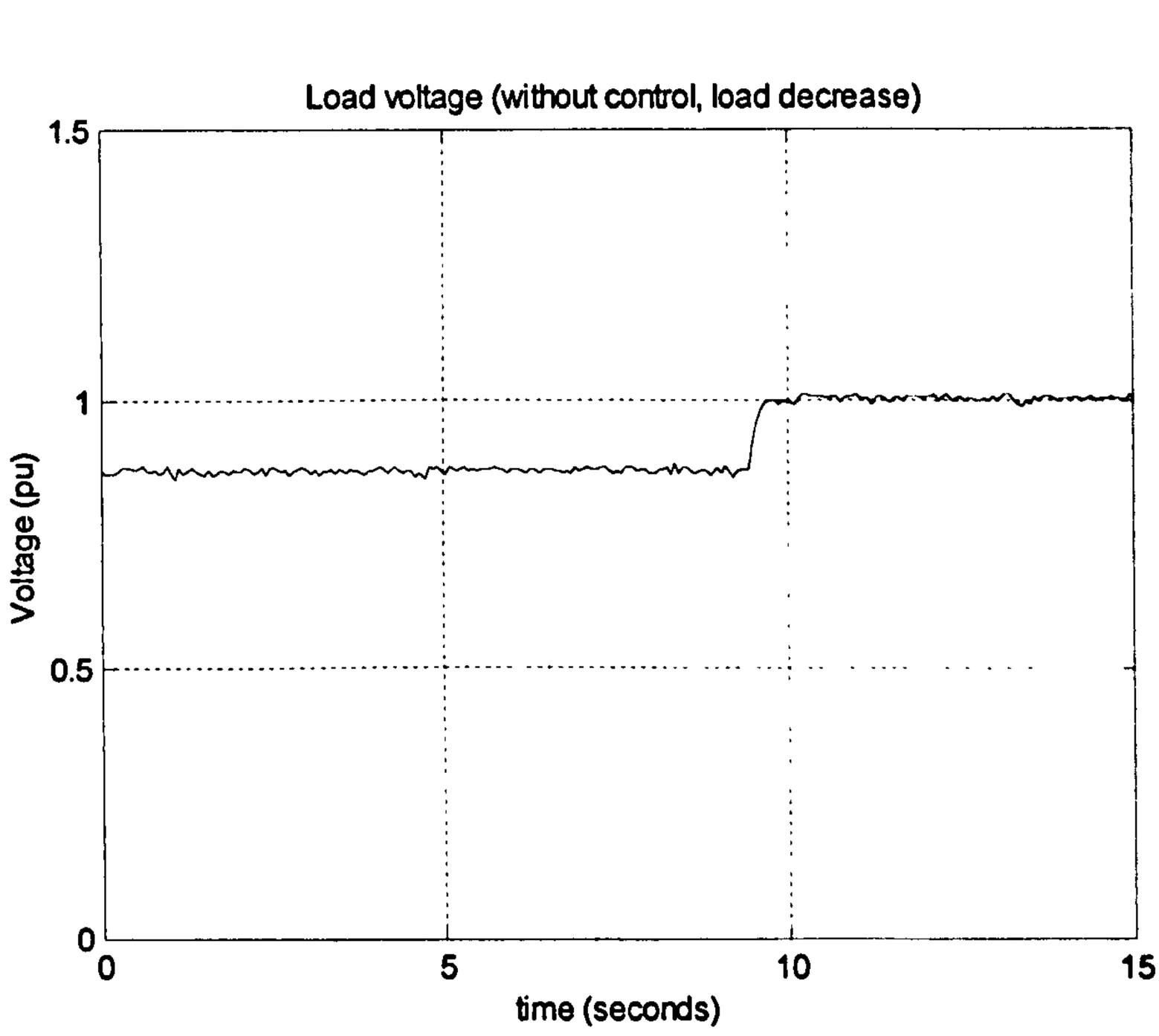


Figure 9.18. (c) DC/DC converter output voltage

Figure 9.18. Experimental results without control (load decrease)

The combined control system with chopper and speed controllers. The objective is to verify that the control system is capable of maintaining the load voltage at a desired level under step loading conditions. In this test, the complete control system is applied to the test system. The tests are conducted under the same load varying conditions as that in the tests without the control systems. Figure 9.19 (a) (b) and (c) show the load current, the machine speed and the load voltage when a load increase takes place in the system. It can be observed that the speed of the system increases (95 % rising to 100%) due to the operation of the control system when the load increases (53% rising to 100%). The results for load decrease (100% down to 55%) are presented in Figure 9.20 (a) (b) and (c), where the system responds to the load decrease by reducing the speed of the system (100% down to 95%). In both cases the load voltage is kept at almost constant. The results show that the proposed control system work well.

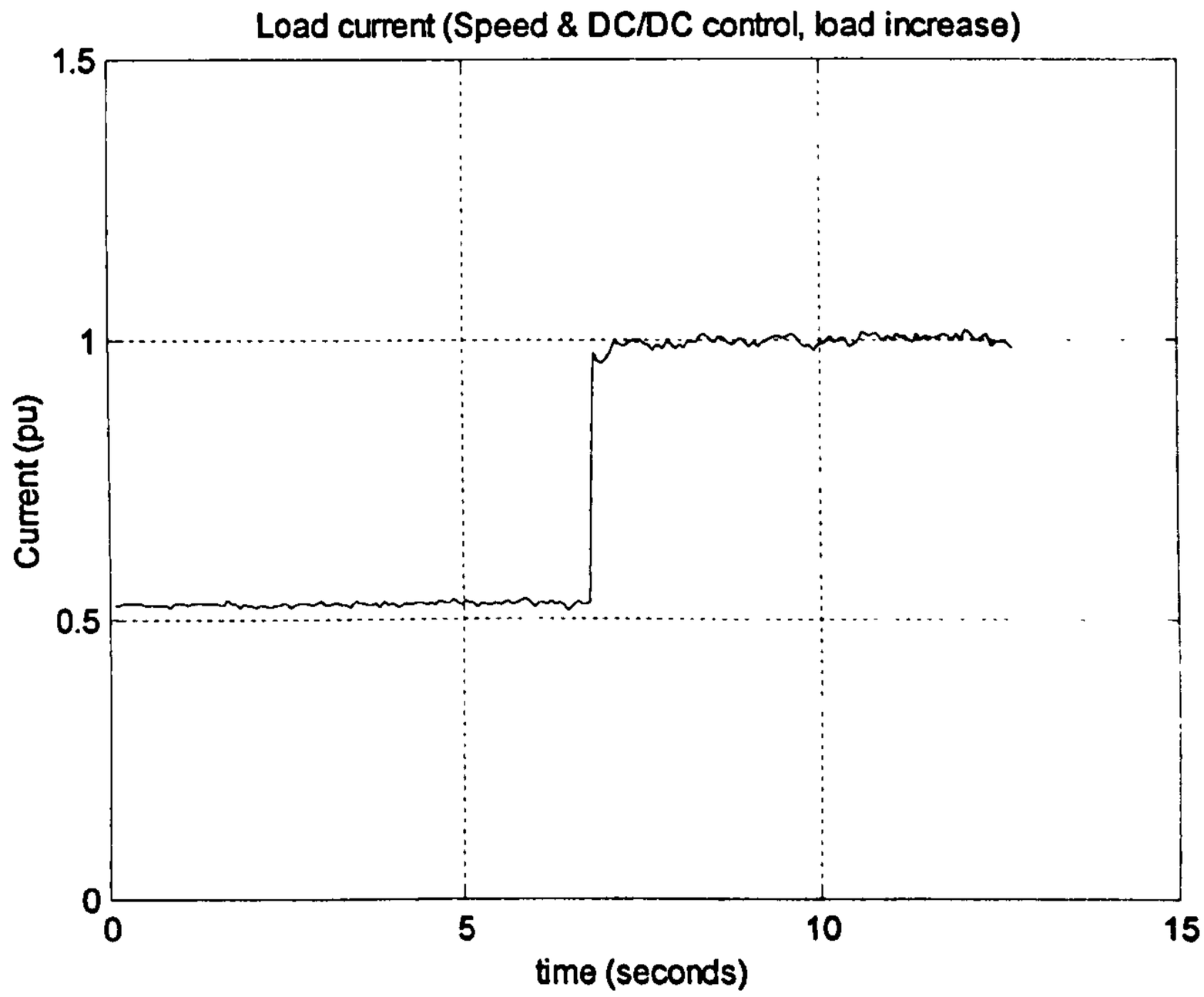


Figure 9.19. (a) Load current

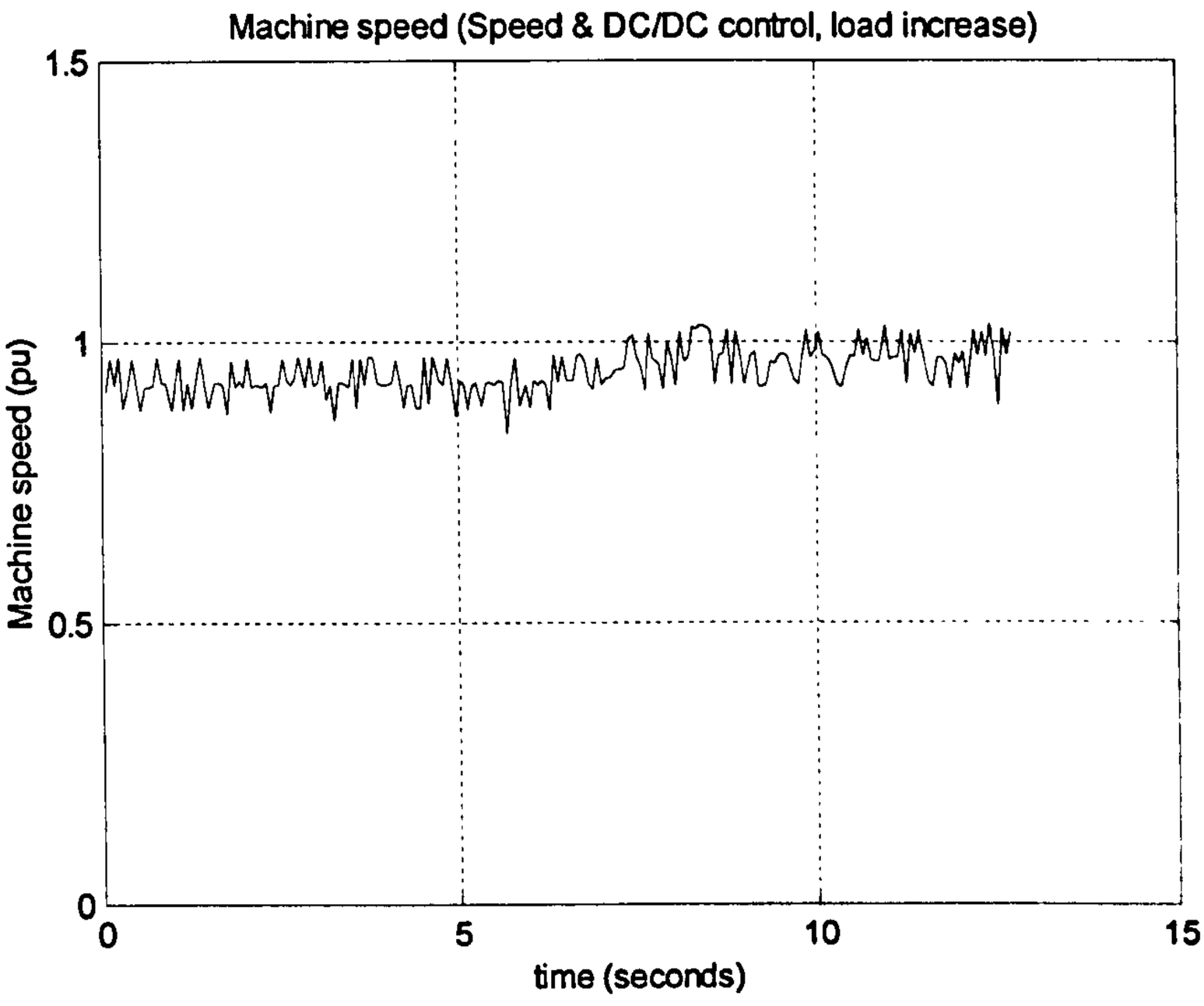


Figure 9.19. (b) Machine speed

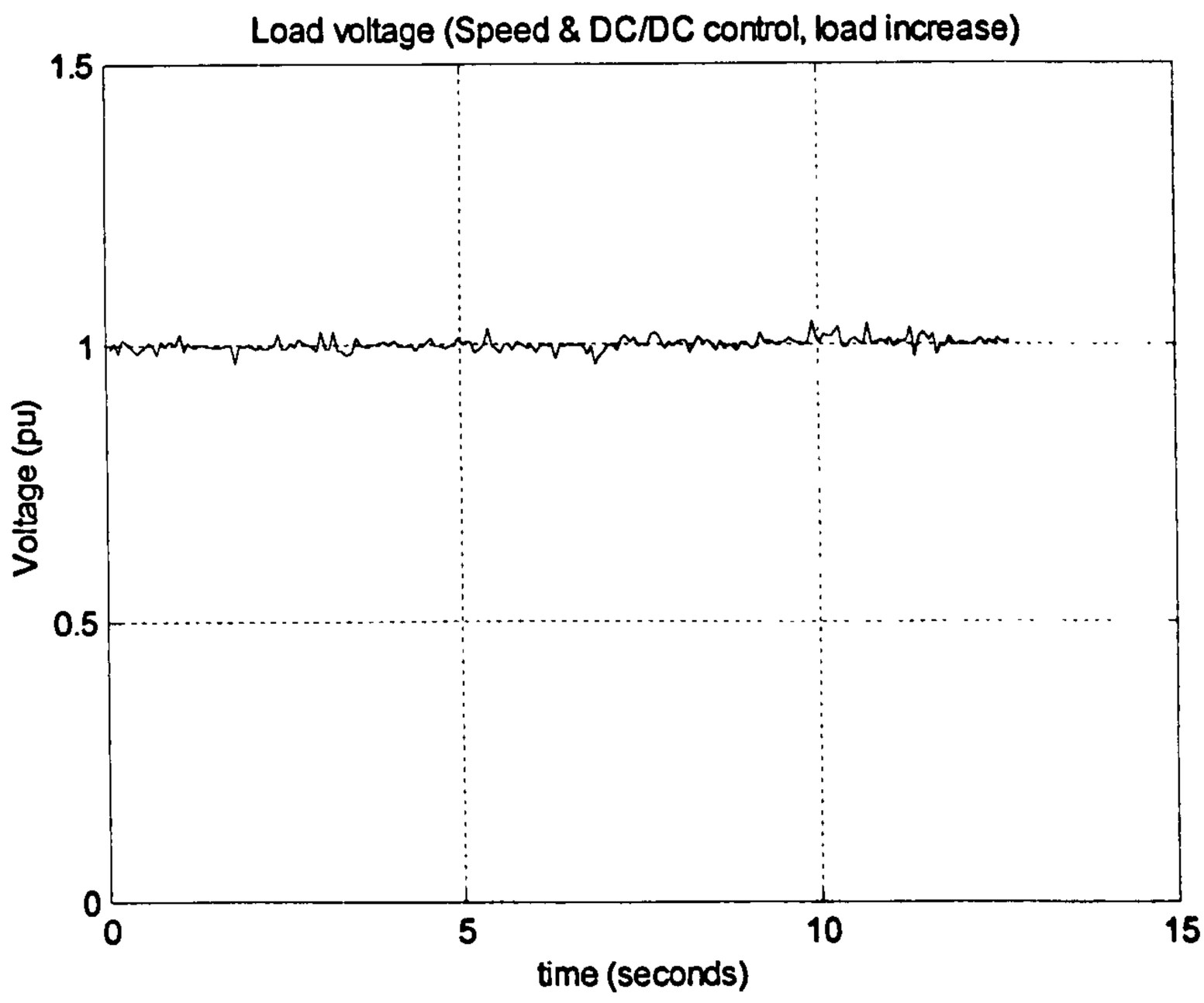


Figure 9.19. (c) DC/DC converter output voltage

Figure 9.19. Experimental results of the combined control system (load increase)

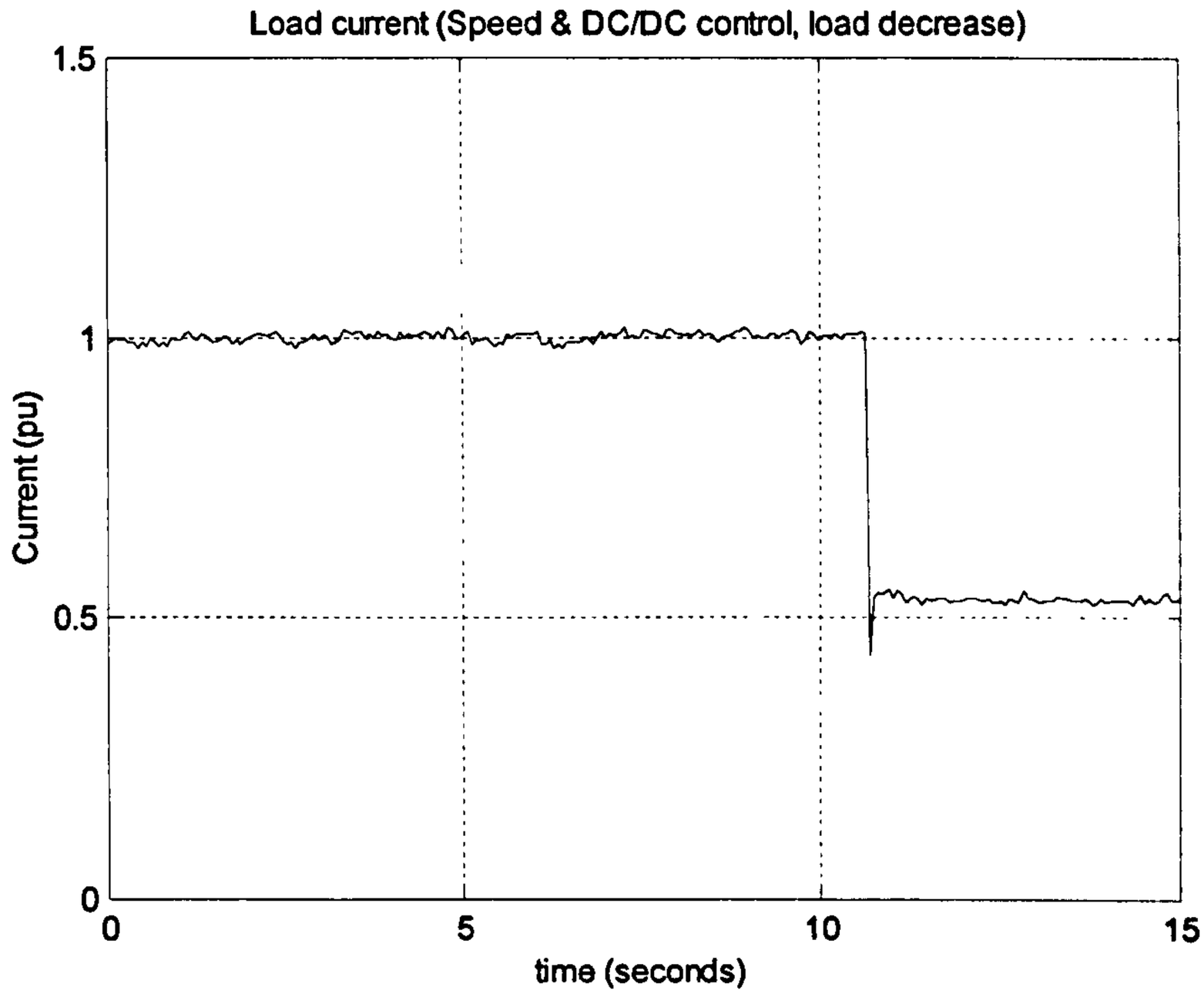


Figure 9.20. (a) Load current

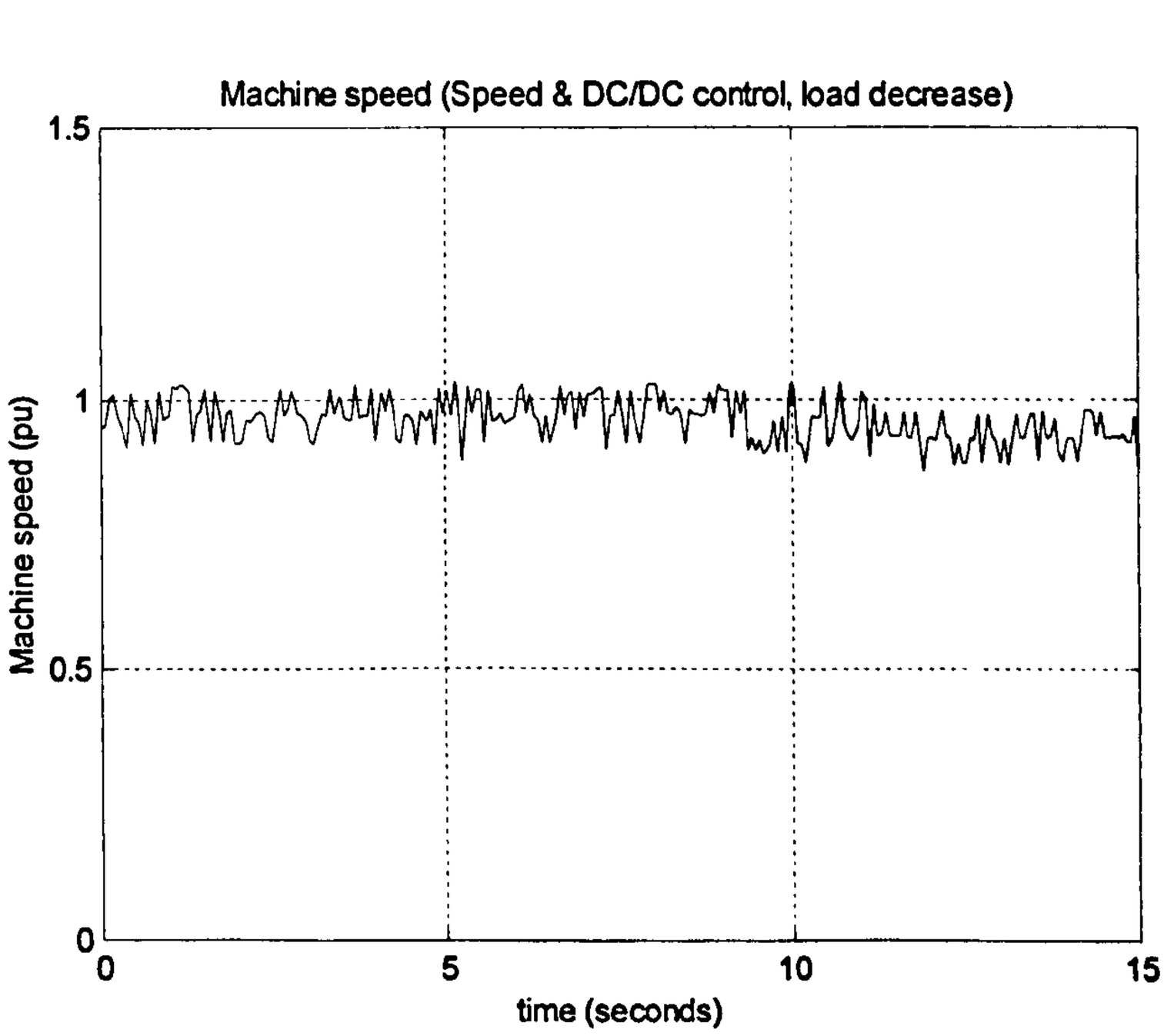


Figure 9.20. (b) Machine speed

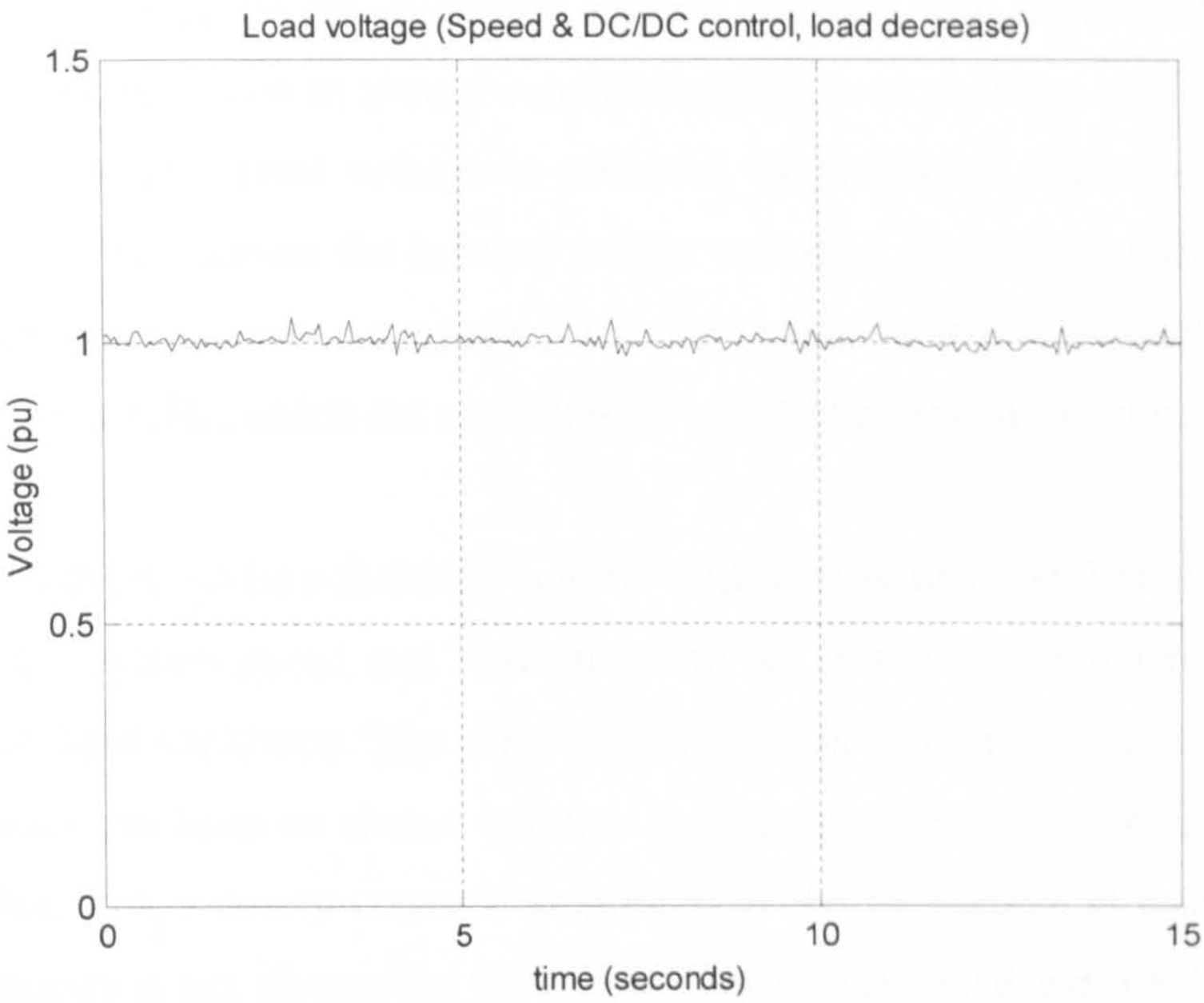


Figure 9.20. (c) DC/DC converter output voltage

Figure 9.20. Experimental results of the combined control system (load decrease)

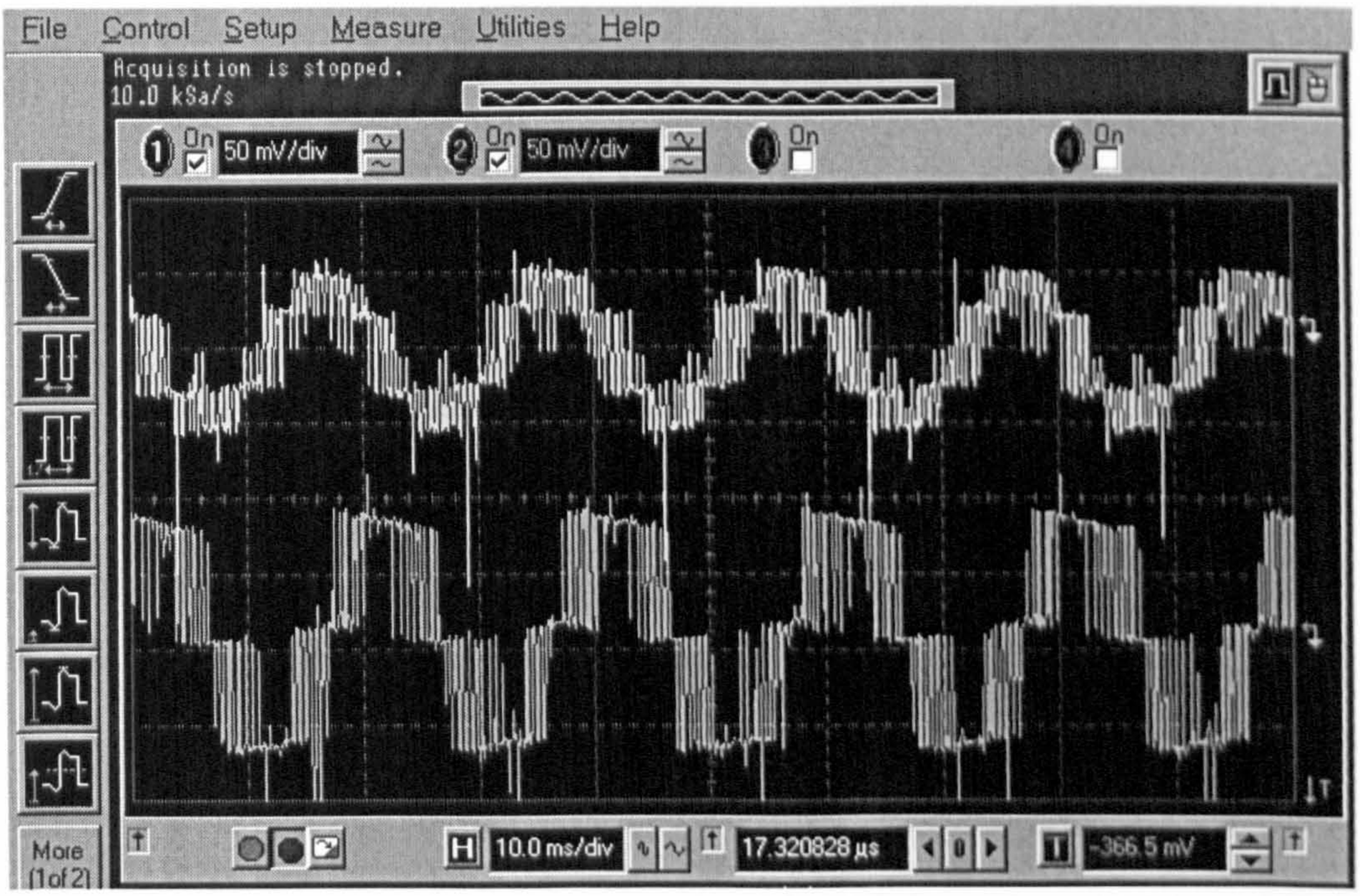


Figure 9.21 Experimental results of voltage source inverter

The experimental study demonstrates that the output voltage of the dc-dc converter can be kept constant by the proposed control system. The inverter with a fixed modulation ratio produces an almost constant amplitude ac voltage, while the constant frequency of the ac output voltage is achieved by timing of the switching control signals. Figure 9.21 shows the inverter output voltages. The upper waveform is the phase voltage and the lower waveform is the line to line voltage. The output frequency of the inverter is 50Hz, and is not significantly affected by speed variations.

From the results, it can be concluded that the controller is successful at simultaneously governing the system speed and regulating the dc voltage level when subjected to large sudden load variations. The experimental studies demonstrate that the proposed control system can keep an almost constant dc voltage at the input dc terminal of the inverter, therefore, a nearly constant amplitude of the ac voltage is ensured. The ac output frequency is not affected by variations in generator speed and the desired output frequency is maintained at all times. This feature allows the variable speed generator set to deliver good quality power to the consumers.

Chapter 10

Conclusions and Recommendation

10.1. Conclusions

This thesis describes research work on the design and implementation of a control system for variable speed stand alone diesel engine driven generators. The main focus has been the development of two advanced controllers for automatic fuel and power electronic converter adjustments to provide quality power to the consumers.

An extensive literature survey on the relevant engineering topics and previous research work on diesel engines, electric machines, power electronic converters and power systems has been carried out. The work has addressed the features of each individual element and the integrated system. The main research developments, achievements and conclusions are summarised in the following sections.

Modelling and simulation of the diesel and generator system

The principal objective was to develop a control system for an isolated diesel-generator system. An indispensable step towards the goal was to develop a system model to enable the investigation and evaluation of appropriate control strategies. The overall system model was particularised by considering real system requirements making the necessary simplifications. Assumptions and simplifications were generally implemented to reduce the calculation burden and satisfy the demands of real-time operation. Simulation models were developed and simulation studies performed in the initial stages of the investigation to establish the validity of the models developed. The main outcomes can be stated as:

- The stand-alone diesel engine driven generator system was divided into subsystems. The operating principles and the features of each subsystem were investigated in detail and the relationship among the subsystems studied. The

subsystem models were all developed in Matlab/Simulink and simulations were carried out respectively in Chapter 3, Chapter 4, Chapter 5 and Chapter 6. These showed that each model predicted the behaviour of the element being studied in an expected manner thereby giving confidence that each developed model is appropriate.

- The diesel engine subsystem model is based on the mathematical equations used to represent the engine performance. Simplifications were applied to reflect practical operating details.
- The generator subsystem is represented using the a-b-c and Park's reference frame. The state equations were transformed to the d-q-0 frame to provide a general excitation machine model. The effects of saturation, eddy currents and demagnetisation are neglected. These effects are important in machine design but in the context of overall system performance are considered to be second order effects and therefore it is considered that they justifiably can be ignored. It is also assumed that a permanent magnet generator can be represented by the above generator model operating with a constant excitation current.

The electromechanical behaviour of the basic system has been investigated using these developed simulation models.

AC/DC/AC power conversion and control system

An AC/DC/AC power electronic conversion system is used for power conversion. In Chapter 5 the space vector PWM techniques used for dc-ac conversion have been thoroughly discussed and the comparison of dualities between CSI and VSI presented. Having considered the voltage requirements of the system, a VSI system was adopted. Simulation studies were carried out to examine the performance characteristics predicted. The simulation studies demonstrated the correctness of the simulation models for the power electronic conversion system and control system. The development of power electronic and control system models has established the foundation for studying the overall system and strategies.

Hybrid variable speed controller

A hybrid variable speed controller was developed to control the speed of the prime mover by adjusting the fuel input as described in Chapter6. A fuzzy logic core was designed for the controller, which responds to the load change and operates the system at a variable speed. The basic characteristics of the controller are that:

- It responds not only to the load change but also to the speed change. For example when the required power changes, the system adjusts its fuel input accordingly to meet the load demand. However both the speed error and the change rate of the speed error affect how the fuel rate is changed during the transient period.
- The developed fuzzy logic method significantly reduces the arithmetic complexity necessary in the model because a highly accurate modelling is not required for decision making.
- The speed of the implemented device does not depend on the number of rules in the membership selection. Furthermore, the simplified on-line reasoning does not need a high capacity and very expensive FPGA chip and hence the cost is reduced.
- The hybrid controller developed comprises both a fuzzy logic core and conventional control techniques. This improves the effectiveness of the controller in dealing with parameter variation and system non-linearities.

The diesel engine generator system exhibits complicated non-linearity. By using the hybrid controller instead of a complicated mathematical model in the design of the variable speed controller, the hardware implementation can be accommodated in a less expensive, compact FPGA device. Consequently a highly robust low cost controller is made available.

System control strategies

To achieve on-line control of the system, a power balance control strategy is proposed in Chapter 7, that is as load increases, the speed controller adjusts the operating speed to a higher level so as to generate more power to balance the load increase.

In the proposed control strategy, keeping a constant dc link voltage is an important condition for providing a constant terminal voltage in the ac system. This is achieved by using a dc-dc converter and a fixed modulation ratio switching PWM inverter. It is shown that the chopper controller is able to respond quickly to the system dynamics and provide fast control of the DC link voltage. The combination of the dc-dc chopper and fixed modulation ratio switching PWM inverter is an effective combination that provides an ability to maintain a constant output voltage under variable load conditions. The gap between high speed system transient response and the moderate speed response of the fuel inlet is bridged by the chopper.

The investigation and evaluation of the control strategy was initially performed by simulation. For all subsystem models simulation results have shown that the respective models work correctly. The combined control strategy for the whole system performs well. The corresponding results demonstrate that the isolated generation system can be controlled to operate at variable speed under the condition of changing load.

The results obtained have proven that the control strategy is effective for stand-alone generation system. The outcome is an effective implementation of the control strategy by the innovative integration of two co-ordinate controllers.

CPLD/FPGA implementation of the control system

The development of the control systems was carried out using Xilinx Foundation Express, a development system which provides computer added design process technology, programmable logic specific high-level flows and optional auto-interactive tools. The development system is used to automatically transform the logic optimisation synthesised design to an effective circuit (the CPLD/FPGA prototyping) and efficiently implement the controllers. The design process is described in Chapter 8 and is briefly summarised.

Using the top-down design methodology, the design starts at a higher-level of abstraction than the traditional gate-level design techniques. The logical or functional abstractions are manipulated using logic synthesis tools for the CPLD/FPGA

implementation. The VHDL code for the complete control system design including the Hybrid Variable Speed Controller and Chopper Controller has been developed. Typical functionality tests have been performed with the simulation tool to verify the CPLD/FPGA design in the form of VHDL description at the early stage in the design process. Static timing analyses are also performed to examine the logic and timing of the implementation and to calculate the performance along signal paths, identify possible race conditions, detect set-up and hold-time violations, which decide the timing suitability for prototyping.

With the design synthesised and analysed for functionality and timing suitability, the controllers are implemented in XC95108-PC84 and XS4010PC84 devices, which are compact and cheap. The developed VHDL code (Appendix B) has in built software control and flexibility and can be adapted to various applications and modified to accommodate other types of control systems. The whole process has been successfully completed and resulted in CPLD and FPGA controllers being available for incorporation in the experimental test rig.

Prototype model construction and experimental studies

An experimental rig based on the F12 MKIV machine system and the power electronic converter systems was constructed. The CPLD and FPGA implemented controllers were integrated into the machine and power electronic systems to carry out speed and voltage regulating tests. Associated measuring circuits and interface electronic circuits were built.

Practical experiments were carried out using the experimental system to study the system performance. A series of experimental tests were carried out for both uncontrolled and controlled conditions when the system was operated under changing load conditions. The developed control system was shown to operate correctly. For example, the simulation results shown in Figure 7.4 are close to the experimental results shown in Figure 9.19 for the case where the system load changes from half load to full load. During this period the load current changed from 0.5 pu to 1.0 pu as shown in Figure 7.4 (a) and Figure 9.19 (a). In response the speed controller (in both

simulation and experiment) adjusted the input power to increase the machine speed (Figure 7.4 (c) and 9.19 (b)) to an appropriate level to achieve power balance. The experimental results are noisy, however, when the signal is averaged it is seen that the calculated speed is very close to the simulation results (machine speed increases from 0.93 pu to 1.0 pu). During the period of change the load voltage is maintained almost constant by the dc/dc controller except for a very short transient period (Figure 7.4 (d) and Figure 9.19 (c)). Clearly, both simulation and experimental studies show that the speed controller appropriately adjusts the input power. In conclusion it can be stated that although the mechanical system responds slowly the co-ordinate control system, including the hybrid variable speed control and the fast dc/dc converter control, successfully maintains a constant load voltage thereby ensuring the required consumer conditions.

10.2. The new contributions of the thesis

A number of innovative contributions have been made and can be summarised as:

- The simulation models of a stand-alone diesel engine driven generator system, including the diesel engine, generator, power electronic system and control system, have been developed in Matlab/Simulink. The performance of the system has been investigated using the developed simulation models.
- A power balance based control strategy proposed for on-line control of the variable speed system to achieve optimal operation.
- A hybrid control method (a combination of classical mathematical method and fuzzy logic method) is developed for the variable speed controller design to achieve a dual-closed-loop control to the diesel engine speed. The hybrid variable speed controller responds to load, speed error and the speed error derivation. The implemented fuzzy logic core does not depend on the number of rules in the membership selection. Therefore, a moderate capacity and low cost FPGA chip can be used. The controller has been implemented in FPGA XS4010PC84 device.

- A CPLD device (XC95108-PC84) implemented voltage controller has been developed for the dc-dc converter to keep a constant dc link voltage for providing an adequate voltage to the consumer in the ac system. The chopper controller can respond quickly to the system dynamics and provide fast control of the DC link voltage.
- The VHDL code for the complete control system design including the hybrid variable speed controller and chopper controller has been developed and has been examined by static timing analyses to decide the timing suitability for prototyping. With the design synthesised and analysed for functionality and timing suitability, the controllers are implemented in compact and cheap XC95108-PC84 and XS4010PC84 devices.
- An overall control strategy has been developed by the integration of two co-ordinate controllers, a variable speed controller and a power electronic dc-dc converter controller. The gap normally evident in system performance when dealing with high speed system transients and the moderate speed response of a mechanical system is effectively bridged by the chopper. The combined control strategy for the whole system performs well.

The contributions made by the research work are in the area of automatic control systems for diesel engine generator sets and CPLD/FPGA application technology. It is anticipated that the work will be of benefit to manufacturers and consequently consumers.

10.3. Recommendation

To date the developed control system has been tested on a small-scale machine set at a fractional horsepower level. In order to fully evaluate the control system, it is recommended that a larger diesel PM generator set to be constructed for industrial appraisal and test.

Publications

[1]. Hu, Y., Cirstea, M.N., McCormick, M., Urwin, Haydock, P., L., "Modelling and Simulation of a Variable Speed Stand-Alone Generator System" Proc. of the IEE Int. Conf. on Power Electronics and Variable Speed Drives PEVD 2000, London, UK, September 2000, pp.372-377.

[2]. Cirstea, M.N., Khor, J.G., Hu, Y., McCormick, M., Haydock, L., "Intelligent Fuzzy Logic Controller for Power Generation Systems" Proc. of the 9th International Conference on Electrical Machines and Drives, Canterbury, UK, 1-3 September 1999, pp.321-324.

[3]. Cirstea, M.N., Khor, J.G., Hu, Y., McCormick, M., Al-Khayat, N., "An Intelligent Fuzzy Logic Controller Implemented in FPGA" Proc. of the International Conference on Power Conversion and Intelligent Motion (PCIM'99), Nurnberg, Germany, 22-24 June 1999, vol. 39, pp. 523-527.

References

[1]. Adkins, B., Harley, R. G., "The General Theory of Alternating Current Machines", Chapman and Hall, London, 1979.

[2]. Armstrong, J. R., "Chip-level modeling with VHDL", Prentice-Hall Inc. 1989.

[3]. Asmus, A., Wellington, B., "Diesel Engine and Fuel Systems", Pitman Publishing PTY Ltd, 1988.

[4]. Blaabjerg, F., Freysson, S., Hanson, H. H., Hanson, S., "A New Optimized Space-Vector Modulation Strategy for a Component-Minimized Voltage Source Inverter", IEEE Trans. on Power Electronics, Vol. 12, No. 4, July 1997, pp. 704-714.

[5]. Bojadziev, G., Bojadziev, M., "Fuzzy Sets, Fuzzy Logic, Applications", Advances in Fuzzy Systems—Applications and Theory, Vol.5. 1995.

[6]. Boost, M. A., Ziogas, P. D., "State-of the Art Carrier PWM Techniques: A Critical Evaluation", IEEE Trans. Ind. Application, Vol. 24, No. 24, Mar/April 1998, pp.271-280.

[7]. Carlson, A., Hylander, J., "Electrical System with Frequency Convertors for Variable Speed Operation of Wind Turbines," Proc. of European Wind Energy Conference and Exhibition EWEC, Peter Peregrinus Ltd., 1989, London, pp.774-778.

[8]. Oldfield, J.V., Dorf, R. C., "Field Programmable Gate Array", John Wiley & Sons, INC, 1995.

[9]. Cavlan, N., Chne, R., "Field-PLAs Simplify Logic Design", Electronic Design, pp.84-90, September 1975.

[10]. Chen, Z., Spooner, E., " Wind Turbine Power Converters: A Comparative Study", IEE International Conference PEVD'98, London, September, 1998, pp. 471-476.

[11]. Chen, Z., Spooner, E., "Voltage Source Inverters for High-Power, Variable-Voltage DC Power Sources", Accepted for IEE Proc. –Generation, Transmission and Distributions.

[12]. Chen, Z., Spooner, E., "Grid Interface Options for Variable-Speed, Permanent-Magnet Generators", IEE Proc. Electr. Power Applications, Vol. 145, No. 4, July 1998, pp.273-283.

[13]. Dinu, A., Cirstea, M., McCormick, M. M., Ometto, A., Rotondale, N., "New Approach for a PWM Inverter Using a State Space Observer", Symposium on Power Electronics, Electrical Drives, Advanced Machines, Power Quality, June 3-5, 1998, Sorrento (Italy), pp.A2-31--A2-36.

[14]. Dinu, A., McCormick, M. M., Cirstea, M., Ometto, A., Rotondale, N., "Neural Network for Control of PWM Inverters", Proceedings of the PEMC'98, Prague, September 8-10, 1998. pp.1-5.

[15]. Dunsheath, P., "A History of Electrical Engineering", Faber and Faber, London, 1962.

[16]. Mohsen, A., "Desktop-Configurable Channelled Gate Arrays", VSLI Systems Design, pp.24-33, August 1988.

[17]. Emms, P. M., "The technical and financial analysis of prime movers", 2nd international conference "Power Generation – The Route to Operating Success", 16 March, 1989.

[18]. Espinoza, J. R., Joós, G., "DSP Implementation of Output Voltage Reconstruction in CSI-Based Converters", IEEE Trans. Ind. Application, Vol. 45, No. 6, Dec. 1998, pp.895-904.

[19]. Habetler, T. G., Divan, D. M., "Acousic Noise Reduction In Sinusoidal PWM Drives Using A Randomly Modulated Carrier", IEEE Trans. on Power Electronics, Vol. 6, July 1991, pp.356-363.

[20]. Haddad, S., Watson, N., "Principle and Performance in Diesel Engineering", Ellis Horwood Limited.

[21]. Hammons, T. J., Winning, D. J., "Comparison of Synchronous-Machine Models in the Study of the Transient Behaviours of Electrical power Systems", Proc. IEE, Vol. 118, No. 10, Oct. 1971, pp.1442-1458.

[22]. Hancock, N. N., "Matrix Analysis of Electrical Machinery", Second Edition, Pergamon Press, 1974.

[23]. Heier, S., "Grid Integration of Wind Energy Conversion Systems", John Wiley & Sons, 1998.

[24]. Holtz, J., Stadtfeld, S., "A Predictive Controller for the Stator Current Vector of A.C. Machines Fed from A Switched Voltage Source", Int. Power Electronics Conf. Tokyo, March 1983, pp.1665-1675.

[25]. Hui, S. Y. R., Oppermann, I., Sathiakumar, S., "Microprocessor-Based Random PWM Schemes for D.C.-A.C. Power Conversion", IEEE Trans. on Power Electronics, Vol. 12, No. 2, Mar 1997. pp.253-260.

[26]. IEEE Committee Report, "Proposed Excitation System Definitions for Synchronous Machines", IEEE Trans, Power App. System, Vo. PAS-88, No. 8, AUG. 1969, pp.1248-1258.

[27]. Judge, A. W., "High Speed Diesel Engines", Chapman & Hall, 1967.

[28]. Jung, S. L., Tzou, Y. Y., "Discrete Sliding Mode Control of A PWM Inverter for Sinusoidal Output Waveform Synthesis with Optimal Sliding Curve", IEEE Trans. on Power Electronics, Vol. 11, No. 4, July 1996, pp.567-577.

[29]. Kerkman, R. J., Rowan, T. M., "Voltage-controlled Current-Regulated PWM Inverters", IEEE Trans. Ind. Application, Vol. 26, No. 2, March/April 1990, pp.244-251.

[30]. Kron, G., "Classification of the Reference Frames of a Synchronous Machine", AIEE Trans. Vol 69, Part II, 1950, pp.720-727.

[31]. Lasnier, F., Ang, T. G., "Photovoltaic Engineering Handbook", IOP Publishing Ltd., 1990.

[32]. Mamdani, E. H., Assilian, S., "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, Vol.7, No.1, pp.1-13, 1975.

[33]. Nabae, A., Ogasawara, S., Akagi, H., "A Novel Control Scheme for Current-Controlled PWM Inverters", IEEE Trans. Ind. Appl. vol IA-22, no. 4, July/August 1986, pp.697-701.

[34]. Novotny, D. W., Lipo, T. A., "Vector Control and dynamics of Drives", Claredon Press Oxford, 1996.

[35]. Oldfield, J. V., Dorf, R. C., "Field Programmable Gate Arrays", John Wiley &| Sons, Inc.1995.

[36]. Onan RV GenSet Operator's Manual and Installation Manual.

[37]. Park, R. H., "Two Reaction Theory of Synchronous Machines", Trans. AIEE, Part I, 48, pp.716-730, 1929 and Part II, 52, pp.352-355, 1933.

[38]. Ross, T. J., "Fuzzy Logic with Engineering Applications", McGraw-Hill, 1995.

[39]. Saadat, H., "Power System Analysis", McGraw-Hill International Editions, 1999.

[40]. Sarma, G., "Power System Analysis& Design", PWS Publishing Company, 1994.

[41]. Stone, R., "Introduction to internal Combustion Engines", Macmillan Press Ltd. 1999.

[42]. Tanaken, "An Introduction to Fuzzy Logic for Practical Applications", Springer.

[43]. The Programmable Gate Array Data Book, Xilinx, Inc., 1999.

[44]. TTL Logic Data Book, Texas Instruments, 1988.

[45]. Tzou, Y. Y., Hsu, H. J., "FPGA Realization of Space-Vector PWM Control IC for Three-Phase PWM Inverters", IEEE Trans. on Power Electronics, vol. 12, No. 6, Nov. 1997, pp.953-963.

[46]. Thorborg, Kjeld, "Power electronics", New York; London, Prentice-Hall, 1988.

[47]. Verdelho, P., Marques, G. D., "Four Wire Active Power Filter Control Circuit with Phase locked Loop Phase Angle Determination", Power electronics and Variable Speed Drives, 21-23 Sept. 1998, Conference Publication No. 456, pp.34-39.

[48]. H., Willi van der Broeck, Hans Cristoph Skudelny, Georg Viktor Stanke, "Analysis and Realisation of a Pulse width Modulator Based on Voltage Space Vectors", IEEE Transactions on Industry Applications, vol. 24, No. 1 January/February 1988, pp.142-149.

[49]. Xiao, Y., Wu, B., DeWinter, F. A., Sotudeh, R., "A Dual GTO Current-Source Converter Topology with Sinusoidal Inputs for High-Power Applications", IEEE Trans. Ind. Application, Vol. 34, No. 4, July/August 1998, pp.878-884.

[50]. Yeager, K. E., Willis, J. R., "Modelling of Emergency Diesel Generators in an 800 Megawatt Nuclear Power Plant", IEEE Trans. On Energy Conversion, Vol. 8, No. 3, Sept. 1993, pp.433-441.

[51]. Zadeh, L. A., "Outline of a new approach to the analysis of complex systems and decision processes", IEEE Trans. On Systems, Man, and Cybernetics, Vol.3, No.1, pp.28-44, Jan. 1973.

[52]. Zelaya de la Parra, H., Iglesias, I. J., Gonzalez, O., Solana, J. M., Tamarit, J., "Modulation and Control of Current Source Converters for High Dynamic Performance of Induction Motors", EPE'97, Trondheim, 1997, Vol. 3, pp.756-761.

Bibliography

[1]. Boost, M.A., Ziogas, P.D., "State-of the Art Carrier PWM Techniques: A Critical Evaluation", IEEE Trans. Ind. Application, Vol. 24, No. 24, Mar/April 1998, pp.271-280.

[2]. Bose, N.K., Garga, A. K., "Neural Network Design Using Voronoi Diagrams" IEEE Transactions on Neural Networks vol. 4, no. 5, September 1993 pp.778-787.

[3]. Chan, P. K., Mourad, S., "Digital Design Using Field Programmable Gate Arrays", Prentice-Hall, 1994.

[4]. Crowder, Richard M.: "Electric Drives and Their Controls", Clarendon Press, Oxford, 1996.

[5]. Driankov, D., Hellendoorn, Reinfrank, H., M., "An Introduction to Fuzzy Control" Springer-Verlag, Berlin Heidelberg 1993.

[6]. Fawcett, B. K., "Taking Advantage of Reconfigurable Logic," The Programmable Gata Array Data Book, Xilinx, Inc. 1992.

[7]. Fodor, D., Vas, J., Katona, Z., "Fuzzy Logic Based Vector Control of AC Motor Using Embedded DSP Controller Board", Proc. of PCIM'98 Intelligent Motion, pp 235-240.

[8]. Garga, Bose A. K., "Neural Network Design Using Voronoi Diagrams", IEEE Transactions on Neural Networks vol. 4, no.5, September 1993, and pp.778-787.

[9]. Khor,J. G., Cirstea, M. N. McCormick, M., Low, W. F., "PWM Control for Variable Speed Stand Alone Generators", Proceedings of the 6th International Conference on Optimization of electrical and Electronic Equipment, Vol. II Power Electronics, Electrical Drives and Automation, Brasov, May 14-15, 1998, pp.379-382.

[10]. Krzeminski, Z., "An Observer System for Induction Motor without Speed Sensor", Proceedings of the 27th International Intelligent Motion Conference, June 1995, Nürnberg Germany, pp.143-153.

[11]. Lasnier, F., Ang, T.G., "Photovoltaic Engineering Handbook", IOP Publishing Ltd., 1990.

[12]. Lin, C. C., Malgorzata, M. S., Gatlin, D., "On Designing Universal Logic Blocks and Their Application to FPGA Design", IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems. Vol.16, No.5, May 1997, pp.519-526.

[13]. Lorenz, R.D., Lawson, B.D., "Performance of Feedforward Current Regulators for Field-Oriented Induction Machine Controllers", IEEE Trans. Ind. Appl. vol IA-23, no. 4, July 1987, pp.597-602.

[14]. Mohan, N., Undeland, T. M., Robbins, W. P.," Power Electronics Converters, Applications, and Design", John Wiley & Sons, 1995.

[15]. Nasar, S.A., "Electric Machines and Power Systems", McGraw-Hill, Inc. 1995.

[16]. Novotny, D.W. and Lipo, T.A. "Vector Control and Dynamics of AC Drives", Clarendon Press Oxford, 1996.

[17]. Perry, D. L., "VHDL", McGraw-Hill Inc. Second edition, 1994. (Computer Engineering Series).

[18]. Rashid, M.H., "Power Electronics", Prentice-Hall International, Inc. 1988.

[19]. Sarma, G., "Power System Analysis& Design", PWS Publishing Company, 1994.

[20]. Sen, P.C., "Principles of Electric Machines and Power Electronics", John Wiley & Sons, Inc. 1997

[21]. Simoes,.E, , "Design and Performance Evaluation of a Fuzzy-Logic-Based Variable Speed Wind Generation System", IEEE Trans. on Inds. Apps., vol. 33, no. 4, 1997, pp.956-965

[22]. Stephenson, G. E. "Power Technology", Litton Educational Publishing, Inc. 1979.

[23]. Takashi Kenjo, "Power Electronics for the Microprocessor Age", Oxford University Press, 1994.

[24]. Urwin, P., Swanick, B.H., "Adaptive Control of Systems with Certain Non-Linear Structures", Int. Journal of Control, Vol.39, No.1, pp.31-55.

[25]. Vithayathil, J., "Power Electronics Principles and Applications", McGraw-Hill, Inc.,1995.

[26]. White, D. A., Sofge, D. A., "Handbook of Intelligent Control Neural, Fuzzy, and Adaptive Approaches" Multiscience Press, Inc. 1992.

[27]. White,D.A., Sofge, D.A., "Handbook of Intelligent Control Neural, Fuzzy, and Adaptive Approaches" Multiscience Press, Inc. 1992.

Appendix A

Development Boards

A.1. XS95-108 board

Features:

- XC95108 CPLD with 108 macrocells
- 8031 microcontroller
- 32 KByte SRAM
- 10 MHz programmable oscillator
- Parallel port
- VGA monitor port
- keyboard/mouse PS/2 port
- 7-segment LED
- 84-pin prototyping interface
- 9V DC power jack
- 5V regulator
- Downloading cable

The XS95-108 Board, shown in Figure A.1, is for experimenting with CPLD designs, microcontroller programming, or hardware/software codesign. The CPLD is used to implement logic circuitry. The microcontroller can use the CPLD as a coprocessor. The 32 KByte SRAM can store microcontroller programs/data or serve as general-purpose storage for CPLD-based designs. The XC9500 series of CPLDs is supported by Xilinx's Foundation Series software. Any recent version of XILINX software should generate SVF configuration files that are compatible with the XS95 Board.

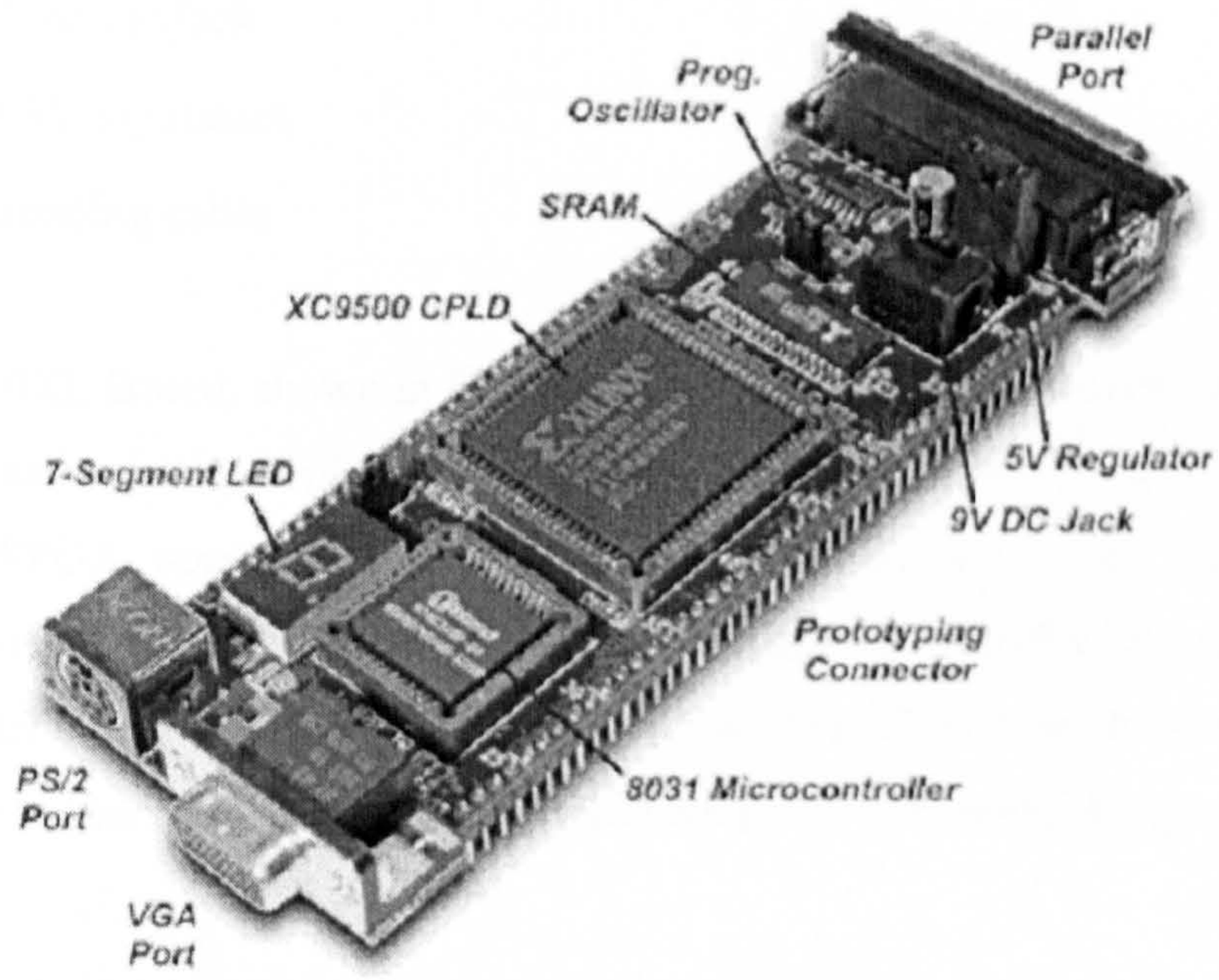


Figure A.1. The XS95-108 Board.

A.2. XS40-010XL board

Features:

- XC4010XL FPGA
- 8031 microcontroller
- 32 KByte SRAM
- 12 MHz programmable oscillator
- Parallel port
- mouse/keyboard PS/2 port
- VGA monitor port
- 7-segment LED
- 84-pin prototyping interface
- Serial EEPROM socket

- 9V DC power jack
- 5V / 3.3V regulators
- Downloading cable

The XS40-010XL Board, shown in Figure A.1, provides a experimental platform for with FPGA designs, microcontroller programming, or hardware/software codesign. The XC4010XL FPGA operates at 3.3V but is 5V-tolerant so it can be connected to commonly available TTL chips. Digital logic designs can be loaded into the FPGA. The microcontroller can use the FPGA as a coprocessor. The SRAM can store microcontroller programs/data or serve as general-purpose storage for FPGA-based designs.

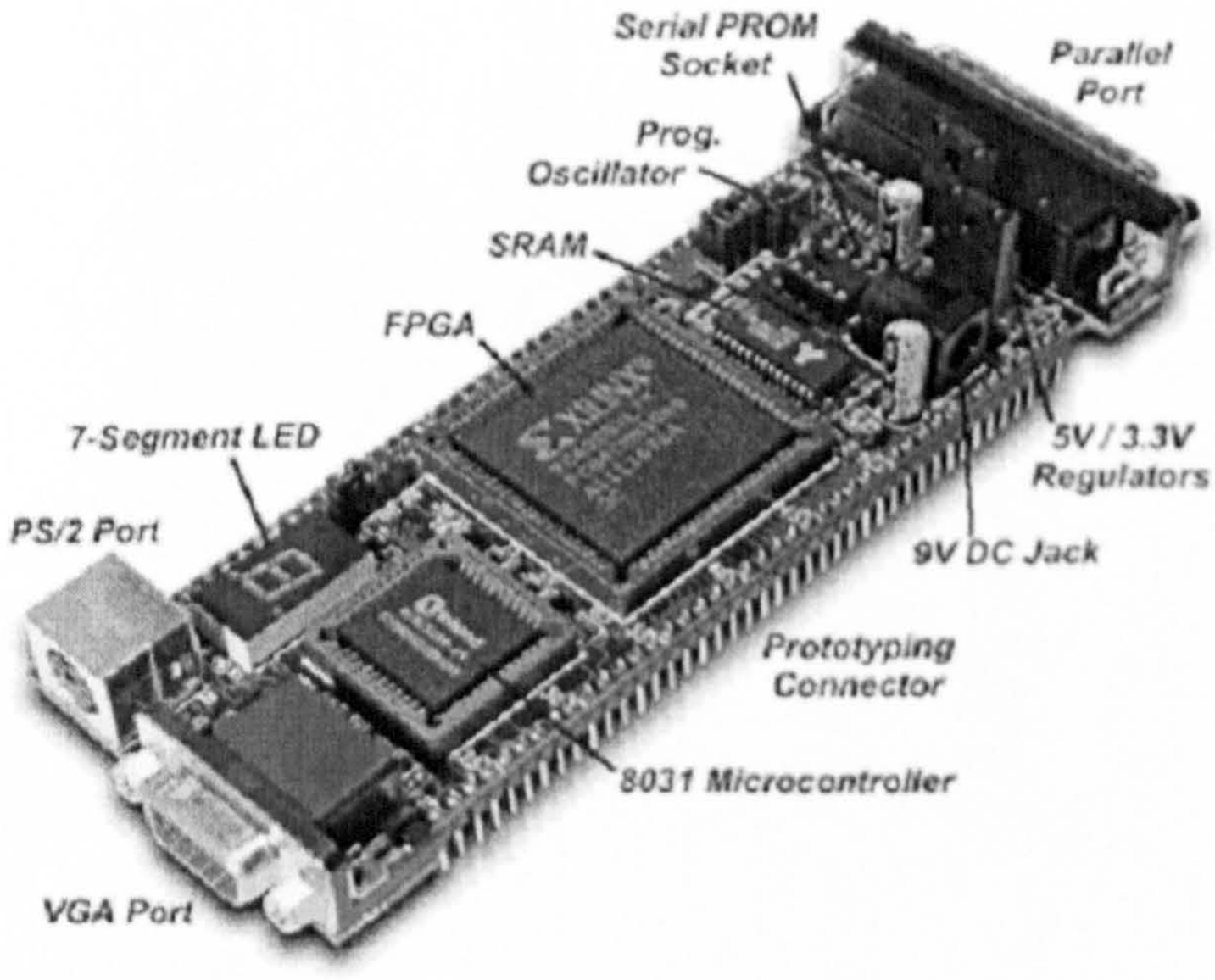


Figure A.2. The XS40-010XL Board

The XC4000XL series of FPGAs is supported by Xilinx's Foundation Series software. Any recent version of XILINX software should generate bitstream configuration files that are compatible with the XS40 Boards.

Y. HU DMU-2001

The two rows of pins from the XS Board can be plugged into a protoboard with holes spaced at 0.1" intervals. Once plugged in, all the pins of the XC4000/XC9500 and the 8031 microcontroller are accessible to other circuits on the protoboard. (The numbers printed next to the rows of pins on the XS Board correspond to the pin numbers of the XC4000 or XC9500.) Power can still be supplied to the XS Board though jack J9, or power can be applied directly through several pins of the XS Board. Just connect +5V and +3.3V to the following VCC pins of the XS40 or XS95 Board, and connect ground to the GND pin.

Appendix B
VHDL Code

B-1.

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_signed.all;
use IEEE.std_logic_arith.all;

entity f0chop95U is
 port (
RESET: in STD_LOGIC;
START: out STD_LOGIC;
CLK: in STD_LOGIC;
Vdc: in STD_LOGIC_VECTOR (7 downto 0);
CdivH6: out std_logic_vector(5 downto 0);
PWMCtrlOut: out STD_LOGIC--_VECTOR (8 downto 0) );
end f0chop95U;

architecture f0chop95U_arch of f0chop95U is
signal int_C :std_logic_vector(12 downto 0);
signal CdivS: STD_LOGIC;
signal CtrlOut: STD_LOGIC;

constant Vref: std_logic_vector(9 downto 0) :="0001111111";
constant CRef: std_logic_vector(9 downto 0) :="0100000000";
signal ctrlVref1: std_logic_vector(8 downto 0);
signal ctrlVref: std_logic_vector(9 downto 0);
signal ctrlCRef: std_logic_vector(9 downto 0);
signal ctrlcount: std_logic_vector(10 downto 0);
signal count: std_logic_vector(8 downto 0);
signal count1: std_logic_vector(8 downto 0);
begin

ctrlVref1<="0"&Vdc;
process (CdivS,RESET)--count,CLK)
variable diff,diff1: std_logic_vector(9 downto 0);

begin
if RESET='1' then
CtrlOut<='0';
elsif CdivS'event and CdivS='1' then
if count1/="000000000" then--and count<=255 then
START<='0';
diff:=Vref-ctrlVref1;
 diff1:=shl(diff,"1");
```

```
--ctrlcount<="0"&(CRef+diff1);
if "0"&(CRef+diff1)<="00000100000" then --Vdc>=EF(hex)
ctrlcount<="00000100000";
elsif "0"&(CRef+diff1)>="00111000000" then--Vdc<=20(hex)
ctrlcount<="00111000000";
else
ctrlcount<="0"&(CRef+diff1);
end if;

if "0"&count<=ctrlcount then
CtrlOut<='1';
else
CtrlOut<='0';
end if;
-- assert (ctrlcount<count) report "ctrlcount<count"
-- severity warning;
else
START<='1';
 end if;--count
 count<=count1+"000000001";
 count1<=count;
 end if;

 end process;
--  ctrlVdc<=Vdc;
process(clk, reset)
begin
if reset='1' then
--    int_C<="0000";
--  elsif clk='1' and clk'event then
--    int_C<=int_C+"0001";
--  end if;
int_C<="0000000000000";
elsif clk='1' and clk'event then
int_C<=int_C+"0000000000001";
end if;
end process;
--CdivS<=int_C(0);
--Cdiv<=int_C;
--PWMCtrlOut<=CtrlOut;

CdivS<=int_C(0);
CdivH6<=int_C(12 downto 7);
PWMCtrlOut<=CtrlOut;

end f0chop95U_arch;
```

B-2.

```
library IEEE;--flclib;
use IEEE.std_logic_1164.all;
use ieee.std_logic_signed.all;
use ieee.std_logic_arith.all;

entity wholectrl_FH2test_L12 is
port (
RST: in STD_LOGIC;
CLK: in STD_LOGIC;
CdivH6: out std_logic_vector(5 downto 0);
Speed: in std_logic_vector(7 downto 0);
Pcur: in STD_LOGIC_VECTOR (7 downto 0);
S_low: out std_logic;
S_high: out std_logic;
abnormal_sig: out std_logic;
READY5: out std_logic;
Uout1: out std_logic_vector(7 downto 0)
);
end wholectrl_FH2test_L12;

architecture wholectrl_FH2test_L12_arch of wholectrl_FH2test_L12 is
function min(L, R: std_logic_vector) return std_logic_vector is
begin
if L < R then
return L;
else
return R;
end if;
end;

function max(L, R: std_logic_vector) return std_logic_vector is
begin
if L < R then
return R;
else
return L;
end if;
end;

signal int_o: std_logic_vector(12 downto 0);
signal Readyin: STD_LOGIC;
signal Sref15: STD_LOGIC_VECTOR (15 downto 0);
signal Sctrlout: STD_LOGIC_VECTOR (7 downto 0);
signal GaindownSref: STD_LOGIC_VECTOR (7 downto 0);
signal CdivS: STD_LOGIC;
signal CtrlOut: STD_LOGIC;
constant Vref: std_logic_vector(9 downto 0) :="0001111111";
constant CRef: std_logic_vector(9 downto 0) :="0100000000";
constant cur0: STD_LOGIC_VECTOR (7 downto 0):="01111111";
constant RaDivKe: STD_LOGIC_VECTOR (7 downto 0):="00010000";
constant RSref0: STD_LOGIC_VECTOR (15 downto 0):="0000101110111000";
signal ctrlVref1: std_logic_vector(8 downto 0);
signal ctrlcount: std_logic_vector(10 downto 0);
```

```
begin
process(clk, RST)
begin
if RST='1' then
int_o<="0000000000000";
elsif clk='1' and clk'event then
int_o<=int_o+"0000000000001";
end if;
end process;
Readyin<=int_o(5);
ctrlVref1<="0"&Vdc;

Process(CLK,RST,Readyin,Pcur)
variable abnormal: std_logic;
variable out_temp: STD_LOGIC_VECTOR (8 downto 0);
variable a0: STD_LOGIC_VECTOR (7 downto 0);
variable b0: STD_LOGIC_VECTOR (7 downto 0);
variable c0: STD_LOGIC_VECTOR (8 downto 0);
variable ivary0: STD_LOGIC_VECTOR (7 downto 0);
variable Smulti0: STD_LOGIC_VECTOR (15 downto 0);--take upper 12 bits
variable Sref0: STD_LOGIC_VECTOR (15 downto 0);
variable Sref01: STD_LOGIC_VECTOR (15 downto 0);
variable c: integer range 0 to 31;
variable time1: std_logic_vector(8 downto 0);
variable time0: std_logic_vector(8 downto 0);
variable diff: std_logic_vector(8 downto 0);
variable Speed_var: std_logic_vector(8 downto 0);
variable Sref_var: std_logic_vector(8 downto 0);
variable error: std_logic_vector(8 downto 0);
constant L0:std_logic_vector(8 downto 0):=conv_std_logic_vector(-60,9);
constant L1:std_logic_vector(8 downto 0):=conv_std_logic_vector(-20,9);
constant L2:std_logic_vector(8 downto 0):="000000000";
constant L3:std_logic_vector(8 downto 0):=conv_std_logic_vector(20,9);
constant L4:std_logic_vector(8 downto 0):=conv_std_logic_vector(60,9);
constant h100: std_logic_vector(8 downto 0):="001100100";
constant z0: std_logic_vector(8 downto 0):="000000000";
variable ADDRA: std_logic_vector(3 downto 0);
variable ADDRB: std_logic_vector(3 downto 0);
variable ALB: std_logic_vector(8 downto 0);
variable BLB: std_logic_vector(8 downto 0);
variable BRA: std_logic_vector(8 downto 0);
variable BRB: std_logic_vector(8 downto 0);
variable tt1: std_logic_vector(8 downto 0);
variable x: std_logic_vector(8 downto 0);
variable ADR: std_logic_vector(1 downto 0);
variable B_A: std_logic_vector(8 downto 0);
variable B_B: std_logic_vector(8 downto 0);
variable temp: std_logic_vector(1 downto 0);
variable temp_A: std_logic_vector(8 downto 0);
variable temp_B: std_logic_vector(8 downto 0);
variable R_sig2: std_logic;
variable c1: std_logic_vector(8 downto 0);
```

```
variable c2: std_logic_vector(8 downto 0);
variable c3: std_logic_vector(8 downto 0);
variable c4: std_logic_vector(8 downto 0);
variable COUNT: std_logic_vector(3 downto 0);
variable xa: std_logic_vector(8 downto 0);
variable xb: std_logic_vector(8 downto 0);
variable xc: std_logic_vector(8 downto 0);
variable ya: std_logic_vector(13 downto 0);
variable yb: std_logic_vector(13 downto 0);
variable yc: std_logic_vector(13 downto 0);
variable za: std_logic_vector(13 downto 0);
variable zb: std_logic_vector(13 downto 0);
variable zc: std_logic_vector(13 downto 0);
variable READY_sig: std_logic;
variable SIGN: std_logic;
variable dp: std_logic_vector(13 downto 0);
variable dn: std_logic_vector(13 downto 0);
variable A: std_logic_vector(13 downto 0);
variable dd1_var: std_logic_vector(12 downto 0);
variable Y1_var: std_logic_vector(12 downto 0);
--variable U_past: std_logic_vector(7 downto 0);
--variable U_var: std_logic_vector(7 downto 0);
variable dcount: integer range 0 to 15;
variable Y: std_logic_vector(7 downto 0);
variable dd1: std_logic_vector(13 downto 0);
variable dd2: std_logic_vector(8 downto 0);
variable win: STD_LOGIC_VECTOR (3 downto 0);
variable S_low0: std_logic;
variable S_high0: std_logic;

begin
if RST='1' then
ivary0:=conv_std_logic_vector(0,8);
Smulti0:="000000000000000";
Sref01:=RSref0;
GaindownSref<="01111111";
Sctrlout<="01111111";
S_low<='0';
S_high<='0';
abnormal_sig<='0';
abnormal:='0';
c:=0;
time1 :="000000000";
time0 :="000000000";

R_sig2:='1';
c1:="000000000";
c2:="000000000";
c3:="000000000";
c4:="000000000";
dd1:="000000000000000";
dd2:="000000001";
--READY_sig:='1';
```

```
COUNT:="0000";
xa:="000000001";
xb:="000000000";
xc:="000000000";
ya:="000000000000000";
yb:="000000000000000";
yc:="000000000000000";
Uout1<="00000000";
READY_sig:='1';
READY5<='0';
dcount:=15;

elsif CLK'event and CLK='1' then
CASE abnormal IS
WHEN '0' =>

CASE Readyin IS
WHEN '0' => c:=c+1;
WHEN '1' => c:=0;
a0:=Pcur;
b0:=cur0;
ivary0:=a0-b0;
Smulti0:=ivary0*RaDivKe;
Sref0:=RSref0+Smulti0;--a0+b0;
Sref15<=Sref0;
Sref01:=shr(Sref0,"100");
GaindownSref<=Sref01(7 downto 0)-60;
S_low<='0';
S_high<='0';
c0:='0'&(Sref01(7 downto 0)-60);
if c0<"000010000" then--lower limite==10000
S_low<='1';
abnormal:='1';
Sctrlout<="00000000";
elsif c0>="010000000" then
S_high<='1';
abnormal:='1';
Sctrlout<="00000000";
else
Sctrlout<=Sref01(7 downto 0)-60;
abnormal:='0';
end if;

WHEN others => c:=0;
END CASE;
case c is
when 0 =>--|1 |2| 3| 4| 5| 6| 7| 8| 9 =>
null;
when 1 =>
if Readyin='0' then
Speed_var(8):='0';
Speed_var(7 downto 0):=Speed;
Sref_var(8):='0';
```

```
Sref_var(7 downto 0):=GaindownSref;
 error:=Speed_var-Sref_var;
time0:=time1;
time1:=error;
diff:= time1-time0;
--x1<=error;
--x2<=diff;
--READY1<='1';
end if;
when 2  =>
x:=error;--x1;
if x<=L0 then
ADR:="00";
B_A:=h100;
B_B:=z0;
elsif (x>L0 and x<=L1) then
ADR:="00";
tt1:=not(x-"01")-"01010";
B_A :=shl(tt1,"1");
B_B:=h100-B_A;

elsif (x>L1 and x<=L2) then
ADR:="01";
tt1:=not(x-"01");
B_A:=shl(tt1,"1")+shl(tt1,"11");
B_B:=h100-B_A;

elsif (x>L2 and x<=L3) then
ADR:="10";
tt1:=shl(x,"1")+shl(x,"11");
B_A:=h100-tt1;
B_B:=tt1;

elsif (x>L3 and x<=L4) then
ADR:="11";
tt1:=x-"01010";
B_B:=shl(tt1,"1");
B_A:=h100-B_B;

else
ADR:="11";
B_A:=z0;
B_B:=h100;
end if;
ADDRA:="00"&ADR;
ALB:=B_A;
BLB:=B_B;
x:=diff;--x2;
if x<=L0 then
ADR:="00";
B_A:=h100;
B_B:=z0;
elsif (x>L0 and x<=L1) then
```

```
ADR:="00";
tt1:=not(x-"01")-"01010";
B_A :=shl(tt1,"1");
B_B:=h100-B_A;

elsif (x>L1 and x<=L2) then
ADR:="01";
tt1:=not(x-"01");
B_A:=shl(tt1,"1")+shl(tt1,"11");
B_B:=h100-B_A;
elsif (x>L2 and x<=L3) then
ADR:="10";
tt1:=shl(x,"1")+shl(x,"11");
B_A:=h100-tt1;
B_B:=tt1;
elsif (x>L3 and x<=L4) then
ADR:="11";
tt1:=x-"01010";
B_B:=shl(tt1,"1");
B_A:=h100-B_B;
else
ADR:="11";
B_A:=z0;
B_B:=h100;
end if;
ADDRB:="00"&ADR;
BRA:=B_A;
BRB:=B_B;
R_sig2:='1';

when 3 =>
win:=ADDRA+ADDRB;
c1:=min(ALB,BRA);
c2:=min(BLB,BRA);
c3:=min(ALB,BRB);
c4:=min(BLB,BRB);
when 4 => --to 11 =>
COUNT:="0000";
xa:=c1;
xb:=max(c2,c3);
xc:=c4;
za:="00000"&xa;
zb:="00000"&xb;
zc:="00000"&xc;
ya:=shl(za,"10");
yb:=shl(zb,"1")+zb;
yc:=shl(zc,"1");

when 5 to 11 =>
if COUNT<=win-1 then
ya:=ya-za;
yb:=yb-zb;
yc:=yc-zc;
```

```
COUNT:=COUNT+1;
end if;

when 12 =>--to 11 =>
dd1:=ya+yb+yc;
dd2:=xa+xb+xc;
--READY4<='1';
COUNT:="0000";

when 13 to 26  =>--25
if  READY_sig='1' then
if dd2="00000000" then
Y1_var:="0000000000000";
READY5<='1';
READY_sig:='1';
else
dp:="00000"&dd2;
dn:=not(dp)+"000000000000001";

SIGN:=dd1(13);
if SIGN='1' then
dd1_var:=not(dd1(12 downto 0))+"01";
else
dd1_var:=dd1(12 downto 0);
end if;

A(13 downto 1):="0000000000000";
A(0):=dd1_var(12);
dd1_var:=shl(dd1_var,"1");
A:=A+dn;
Y1_var(12):=not(A(13));
dcount:=15;
end if;
elsif (READY_sig='0' and dcount>2) then
dcount:=dcount-1;
A:=shl(A,"1");
A(0):=dd1_var(12);
dd1_var:=shl(dd1_var,"1");

if Y1_var(dcount+1)='0' then
A:=A+dp;
else
A:=A+dn;
end if;
Y1_var(dcount):=not(A(13));

end if;

if dcount=2 then
dcount:=dcount-1;
end if;

READY5<='0';
```

```
READY_sig:='0';

when 27  =>
Y:=Y1_var(7 downto 0);

when 28  =>
if abnormal/='1' then
if SIGN='1' then
out_temp:='0'&(GaindownSref-Y);
--end if;
else
out_temp:='0'&(GaindownSref+Y);
end if;

out_temp:='0'&(not(out_temp(7 downto 0)));
out_temp:=shr(out_temp, "1");
Uout1<=out_temp(7 downto 0);

READY5<='1';
READY_sig:='1';

else
Uout1<="00000000";
end if;
when others =>
null;
            end case;
            WHEN '1' =>
            Uout1<="00000000";
            abnormal_sig<='1';
            WHEN others =>
            null;
            end case;
            end if;
            end Process;
end wholectrl_FH2test_L12_arch;
```

Appendix C

Implemented Equations

```
XACT:  version C.16                              Xilinx Inc.
                                  Fitter Report
   Design Name: chopxc95
Fitting Status: Successful                        Date:  6-27-2001, 12:14PM

******************************  Resource Summary  ******************************

Design          Device              Macrocells      Product Terms   Pins
Name            Used                Used            Used            Used
chopxc95        XC95108-7-PC84      84 /108  ( 77%) 480/540  ( 88%) 16 /69   ( 23%)

PIN RESOURCES:

Signal Type     Required     Mapped  |  Pin Type            Used   Remaining
------------------------------------|-------------------------------------
Input        :     8           8    |  I/O              :    14       49

Output       :     6           6    |  GCK/IO           :     1        2
Bidirectional:     0           0    |  GTS/IO           :     0        2
GCK          :     1           1    |  GSR/IO           :     1        0
GTS          :     0           0    |
GSR          :     1           1    |
                 ----        ----
       Total      16          16

;;-------------------------------------------------------------------;;
; Implemented Equations.

 "$OpTx$$OpTx$FX_DC$16_INV$952"  =  ;Imported pterms FB1_7
         "ctrlcount<6>" * /"count<6>".LFBK

 "$OpTx$$OpTx$FX_DC$17_INV$953"  =  "ctrlcount<5>" * /"count<5>".LFBK

 "$OpTx$$OpTx$FX_DC$22_INV$951"  =  /"count<7>" * "ctrlcount<7>"

 "$OpTx$$OpTx$FX_DC$45_INV$950"  =  ;Imported pterms FB2_15
         /"count1<1>" * /"count1<0>" * FC_3_.OUT

 "$OpTx$$OpTx$FX_SC$49_INV$954"  =  RESET
       + /"diff1<8>".LFBK * "diff1<10>".LFBK
       + /"diff1<9>".LFBK * "diff1<10>".LFBK
       + "diff1<8>".LFBK * "diff1<9>".LFBK *
       /"diff1<10>".LFBK
;Imported pterms FB4_15
       + /"count1<1>" * /"count1<0>" * FC_3_.OUT
       + /"diff1<1>" * "diff1<8>".LFBK * "diff1<9>".LFBK *
       /"diff1<6>".LFBK * /"diff1<7>".LFBK * /"diff1<4>".LFBK *
       /"diff1<3>".LFBK * /"diff1<5>".LFBK
       + /"diff1<2>" * "diff1<8>".LFBK * "diff1<9>".LFBK *
       /"diff1<6>".LFBK * /"diff1<7>".LFBK * /"diff1<4>".LFBK *
       /"diff1<3>".LFBK * /"diff1<5>".LFBK

"$OpTx$$OpTx$FX_SC$50_INV$956"  =  RESET
       + /"count1<1>" * /"count1<0>" * FC_3_.OUT
       + "diff1<8>".LFBK * "diff1<9>".LFBK *
       "diff1<6>".LFBK * "diff1<10>".LFBK
       + "diff1<8>".LFBK * "diff1<9>".LFBK *
```

```
        "diff1<7>".LFBK * "diff1<10>".LFBK
        + "diff1<8>".LFBK * "diff1<9>".LFBK *
        "diff1<3>".LFBK * "diff1<10>".LFBK
;Imported pterms FB4_4
        + "diff1<8>".LFBK * "diff1<9>".LFBK *
        "diff1<4>".LFBK * "diff1<10>".LFBK
;Imported pterms FB4_6
        + "diff1<8>".LFBK * "diff1<9>".LFBK *
        "diff1<5>".LFBK * "diff1<10>".LFBK
        + /"diff1<8>".LFBK * /"diff1<9>".LFBK *
        /"diff1<6>".LFBK * /"diff1<10>".LFBK
        + /"diff1<8>".LFBK * /"diff1<9>".LFBK *
        /"diff1<7>".LFBK * /"diff1<10>".LFBK
        + "diff1<1>" * "diff1<2>" * "diff1<8>".LFBK *
        "diff1<9>".LFBK * "diff1<10>".LFBK

 "$OpTx$$OpTx$FX_SC$51_INV$955"  =  RESET
        + "diff1<10>".LFBK
;Imported pterms FB4_14
        + "diff1<8>".LFBK * "diff1<9>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
        + /"diff1<8>".LFBK * /"diff1<9>".LFBK *
        /"diff1<6>".LFBK
        + /"diff1<8>".LFBK * /"diff1<9>".LFBK *
        /"diff1<7>".LFBK

 "$OpTx$FX_DC$37"  =  "count<1>" * /"ctrlcount<1>"
        + "count<1>" * "count<0>" * /"ctrlcount<0>".LFBK
        + "count<0>" * /"ctrlcount<1>" *
        /"ctrlcount<0>".LFBK

 "$OpTx$FX_SC$39"  =  "count<8>" * /"ctrlcount<9>"
        + /"ctrlcount<8>" * /"ctrlcount<9>"

 "$OpTx$syn858/syn858_D2_INV$949"  =  RESET
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT

 "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"  =  "diff1<3>"
        Xor "ctrlcount1<2>" * "diff1<2>"
        + "ctrlcount1<1>" * "ctrlcount1<2>" * "diff1<1>"
        + "ctrlcount1<1>" * "ctrlcount1<2>" * FC_2_.OUT
;Imported pterms FB1_1
        + "ctrlcount1<1>" * "diff1<1>" * "diff1<2>"
        + "ctrlcount1<1>" * "diff1<2>" * FC_2_.OUT
        + "ctrlcount1<2>" * "diff1<1>" * FC_2_.OUT
        + "diff1<1>" * "diff1<2>" * FC_2_.OUT

 "C128_C2_C6/C128_C2_C6_D2"  =  "ctrlcount1<2>".LFBK * "diff1<2>".LFBK
        + "ctrlcount1<1>".LFBK * "ctrlcount1<2>".LFBK *
        "diff1<1>".LFBK
        + "ctrlcount1<0>" * "diff1<0>" *
        "ctrlcount1<1>".LFBK * "ctrlcount1<2>".LFBK
        + "ctrlcount1<0>" * "diff1<0>" *
        "ctrlcount1<2>".LFBK * "diff1<1>".LFBK
;Imported pterms FB3_13
        + "ctrlcount1<1>".LFBK * "diff1<2>".LFBK *
        "diff1<1>".LFBK
        + "ctrlcount1<0>" * "diff1<0>" * "diff1<2>".LFBK *
        "diff1<1>".LFBK

 "C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D"  =  "diff1<4>".LFBK
        Xor "ctrlcount1<3>" * "diff1<3>".LFBK
        + "ctrlcount1<2>" * "ctrlcount1<3>" * "diff1<2>"
        + "ctrlcount1<2>" * "diff1<2>" * "diff1<3>".LFBK
        + "ctrlcount1<1>" * "ctrlcount1<2>" *
        "ctrlcount1<3>" * "diff1<1>"
```

```
;Imported pterms FB4_1
        + "ctrlcount1<1>" * "ctrlcount1<3>" * "diff1<2>" *
        FC_2_.OUT
        + "ctrlcount1<1>" * "diff1<1>" * "diff1<2>" *
        "diff1<3>".LFBK
        + "ctrlcount1<1>" * "diff1<2>" * FC_2_.OUT *
        "diff1<3>".LFBK
        + "ctrlcount1<2>" * "ctrlcount1<3>" * "diff1<1>" *
        FC_2_.OUT
        + "ctrlcount1<2>" * "diff1<1>" * FC_2_.OUT *
        "diff1<3>".LFBK
;Imported pterms FB4_2
        + "ctrlcount1<3>" * "diff1<1>" * "diff1<2>" *
        FC_2_.OUT
        + "diff1<1>" * "diff1<2>" * FC_2_.OUT *
        "diff1<3>".LFBK
;Imported pterms FB4_17
        + "ctrlcount1<1>" * "ctrlcount1<2>" *
        "ctrlcount1<3>" * FC_2_.OUT
        + "ctrlcount1<1>" * "ctrlcount1<2>" * "diff1<1>" *
        "diff1<3>".LFBK
        + "ctrlcount1<1>" * "ctrlcount1<2>" * FC_2_.OUT *
        "diff1<3>".LFBK
        + "ctrlcount1<1>" * "ctrlcount1<3>" * "diff1<1>" *
        "diff1<2>"

 "C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D"  =  "diff1<5>"
        Xor "ctrlcount1<4>" * "diff1<4>"
        + "ctrlcount1<4>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "ctrlcount1<3>".LFBK
        + "diff1<4>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "ctrlcount1<3>".LFBK
        + "ctrlcount1<2>" * "diff1<2>" * "diff1<4>" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
;Imported pterms FB2_4
        + "ctrlcount1<2>" * "ctrlcount1<4>" * "diff1<2>" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
        + "ctrlcount1<1>" * "ctrlcount1<2>" *
        "ctrlcount1<4>" * "diff1<1>" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
        + "ctrlcount1<1>" * "ctrlcount1<2>" * "diff1<1>" *
        "diff1<4>" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
        + "ctrlcount1<1>" * "ctrlcount1<4>" * "diff1<1>" *
        "diff1<2>" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
        + "ctrlcount1<1>" * "diff1<1>" * "diff1<2>" *
        "diff1<4>" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
;Imported pterms FB2_3
        + "ctrlcount1<1>" * "ctrlcount1<2>" *
        "ctrlcount1<4>" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "ctrlcount1<0>".LFBK * "diff1<0>".LFBK
        + "ctrlcount1<1>" * "ctrlcount1<4>" * "diff1<2>" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "ctrlcount1<0>".LFBK *
"diff1<0>".LFBK
        + "ctrlcount1<2>" * "ctrlcount1<4>" * "diff1<1>" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "ctrlcount1<0>".LFBK *
"diff1<0>".LFBK
        + "ctrlcount1<4>" * "diff1<1>" * "diff1<2>" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "ctrlcount1<0>".LFBK *
"diff1<0>".LFBK
;Imported pterms FB2_6
        + "ctrlcount1<1>" * "ctrlcount1<2>" * "diff1<4>" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "ctrlcount1<0>".LFBK *
"diff1<0>".LFBK
        + "ctrlcount1<1>" * "diff1<2>" * "diff1<4>" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "ctrlcount1<0>".LFBK *
"diff1<0>".LFBK
        + "ctrlcount1<2>" * "diff1<1>" * "diff1<4>" *
```

```
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "ctrlcount1<0>".LFBK *
"diff1<0>".LFBK
        + "diff1<1>" * "diff1<2>" * "diff1<4>" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "ctrlcount1<0>".LFBK *
"diff1<0>".LFBK

 "C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D"  =  "diff1<6>"
        Xor "ctrlcount1<4>" * "ctrlcount1<5>" * "diff1<4>"
        + "ctrlcount1<4>" * "ctrlcount1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
        + "ctrlcount1<5>" * "diff1<4>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
;Imported pterms FB6_4
        + "ctrlcount1<5>" * "diff1<5>"
        + "ctrlcount1<4>" * "diff1<4>" * "diff1<5>"
        + "ctrlcount1<3>" * "ctrlcount1<4>" *
        "ctrlcount1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
        + "ctrlcount1<3>" * "ctrlcount1<5>" * "diff1<4>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
        + "ctrlcount1<4>" * "diff1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
;Imported pterms FB6_3
        + "ctrlcount1<3>" * "ctrlcount1<4>" * "diff1<5>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
        + "ctrlcount1<3>" * "diff1<4>" * "diff1<5>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"
        + "diff1<4>" * "diff1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D"

 "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D"  =  "diff1<8>"
        Xor "C128_C7_C6/C128_C7_C6_D215_".LFBK

 "C128_C7_C6/C128_C7_C6_D2"  =  "ctrlcount1<3>" * "ctrlcount1<4>" *
        "ctrlcount1<5>" * "diff1<7>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<4>" * "ctrlcount1<5>" *
        "ctrlcount1<7>" * "C128_C2_C6/C128_C2_C6_D2" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<4>" * "ctrlcount1<5>" * "diff1<7>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<5>" * "ctrlcount1<7>" * "diff1<4>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<5>" * "diff1<7>" * "diff1<4>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
;Imported pterms FB6_16
        + "ctrlcount1<5>" * "ctrlcount1<7>" * "diff1<5>" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<5>" * "diff1<7>" * "diff1<5>" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<3>" * "ctrlcount1<4>" *
        "ctrlcount1<5>" * "ctrlcount1<7>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<3>" * "ctrlcount1<5>" *
        "ctrlcount1<7>" * "diff1<4>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<3>" * "ctrlcount1<5>" * "diff1<7>" *
        "diff1<4>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
;Imported pterms FB6_15
        + "ctrlcount1<4>" * "ctrlcount1<7>" * "diff1<4>" *
        "diff1<5>" *
```

```
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<4>" * "diff1<7>" * "diff1<4>" *
"diff1<5>" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<3>" * "ctrlcount1<4>" *
"ctrlcount1<7>" * "diff1<5>" *
"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<3>" * "ctrlcount1<4>" * "diff1<7>" *
"diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<3>" * "diff1<7>" * "diff1<4>" *
"diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
;Imported pterms FB6_18
+ "ctrlcount1<7>" * "diff1<7>"
+ "ctrlcount1<6>" * "ctrlcount1<7>" *
"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<6>" * "diff1<7>" *
"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<4>" * "ctrlcount1<5>" *
"ctrlcount1<7>" * "diff1<4>" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<4>" * "ctrlcount1<5>" * "diff1<7>" *
"diff1<4>" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
;Imported pterms FB6_1
+ "ctrlcount1<3>" * "ctrlcount1<7>" * "diff1<4>" *
"diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<4>" * "ctrlcount1<7>" * "diff1<5>" *
"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<4>" * "diff1<7>" * "diff1<5>" *
"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<7>" * "diff1<4>" * "diff1<5>" *
"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "diff1<7>" * "diff1<4>" * "diff1<5>" *
"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK

"C128_C7_C6/C128_C7_C6_D215_" = "ctrlcount1<3>" * "ctrlcount1<4>" *
"ctrlcount1<5>" * "diff1<7>" *
"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<4>" * "ctrlcount1<5>" *
"ctrlcount1<7>" * "C128_C2_C6/C128_C2_C6_D2" *
/"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<4>" * "ctrlcount1<5>" * "diff1<7>" *
"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<5>" * "ctrlcount1<7>" * "diff1<4>" *
"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<5>" * "diff1<7>" * "diff1<4>" *
"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
;Imported pterms FB6_11
+ "ctrlcount1<5>" * "ctrlcount1<7>" * "diff1<5>" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<5>" * "diff1<7>" * "diff1<5>" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
+ "ctrlcount1<3>" * "ctrlcount1<4>" *
"ctrlcount1<5>" * "ctrlcount1<7>" *
"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
```

```
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<3>" * "ctrlcount1<5>" *
        "ctrlcount1<7>" * "diff1<4>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<3>" * "ctrlcount1<5>" * "diff1<7>" *
        "diff1<4>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
;Imported pterms FB6_10
        + "ctrlcount1<4>" * "ctrlcount1<7>" * "diff1<4>" *
        "diff1<5>" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<4>" * "diff1<7>" * "diff1<4>" *
        "diff1<5>" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<3>" * "ctrlcount1<4>" *
        "ctrlcount1<7>" * "diff1<5>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<3>" * "ctrlcount1<4>" * "diff1<7>" *
        "diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<3>" * "diff1<7>" * "diff1<4>" *
        "diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
;Imported pterms FB6_13
        + "ctrlcount1<7>" * "diff1<7>"
        + "ctrlcount1<6>" * "ctrlcount1<7>" *
        "C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<6>" * "diff1<7>" *
        "C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<4>" * "ctrlcount1<5>" *
        "ctrlcount1<7>" * "diff1<4>" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<4>" * "ctrlcount1<5>" * "diff1<7>" *
        "diff1<4>" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
;Imported pterms FB6_14
        + "ctrlcount1<3>" * "ctrlcount1<7>" * "diff1<4>" *
        "diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<4>" * "ctrlcount1<7>" * "diff1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<4>" * "diff1<7>" * "diff1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "ctrlcount1<7>" * "diff1<4>" * "diff1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK
        + "diff1<7>" * "diff1<4>" * "diff1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D".LFBK

FC_0_   =  /"$OpTx$$OpTx$FX_DC$22_INV$951" * "$OpTx$FX_SC$39"        ; FC node

FC_1_   =  "count1<1>" * "count1<0>"   ; FC node

FC_2_   =  "ctrlcount1<0>" * "diff1<0>"        ; FC node

FC_3_   =  /"count1<2>" * /"count1<3>" * /"count1<5>" *
           /"count1<4>" * /"count1<6>" * /"count1<7>" * /"count1<8>"    ; FC node

 "Cdiv<0>"  :=  ;Imported pterms FB3_3
          /"Cdiv<0>".LFBK
    "Cdiv<0>".CLKF  =  CLK    ;FCLK/GCK
    "Cdiv<0>".RSTF  =  RESET  ;GSR
    "Cdiv<0>".PRLD  =  GND
```

```
"Cdiv<1>".T  =  "Cdiv<0>".LFBK
   "Cdiv<1>".CLKF  =  CLK    ;FCLK/GCK
   "Cdiv<1>".RSTF  =  RESET  ;GSR
   "Cdiv<1>".PRLD  =  GND

"Cdiv<2>".T  =  "Cdiv<0>".LFBK * "Cdiv<1>".LFBK
   "Cdiv<2>".CLKF  =  CLK    ;FCLK/GCK
   "Cdiv<2>".RSTF  =  RESET  ;GSR
   "Cdiv<2>".PRLD  =  GND

 "Cdiv<3>".T  =  "Cdiv<0>".LFBK * "Cdiv<1>".LFBK * "Cdiv<2>".LFBK
   "Cdiv<3>".CLKF  =  CLK    ;FCLK/GCK
   "Cdiv<3>".RSTF  =  RESET  ;GSR
   "Cdiv<3>".PRLD  =  GND

/PWMCtrlOut  :=  "ctrlcount<10>" * /"$OpTx$$OpTx$FX_DC$45_INV$950"
       + "$OpTx$$OpTx$FX_DC$45_INV$950" * /PWMCtrlOut.LFBK
       + "count<8>" * /"ctrlcount<8>" * /"ctrlcount<9>" *
       /"$OpTx$$OpTx$FX_DC$45_INV$950"
       + /"ctrlcount<6>" * /"$OpTx$$OpTx$FX_DC$45_INV$950" *
       FC_0_.OUT * "count<6>".LFBK
       + /"ctrlcount<5>" * /"$OpTx$$OpTx$FX_DC$45_INV$950" *
       FC_0_.OUT * "count<5>".LFBK *
       /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK
;Imported pterms FB1_4
       + "count<7>" * "count<8>" * /"ctrlcount<7>" *
       /"ctrlcount<9>" * /"$OpTx$$OpTx$FX_DC$45_INV$950"
       + "count<7>" * /"ctrlcount<7>" * /"ctrlcount<8>" *
       /"ctrlcount<9>" * /"$OpTx$$OpTx$FX_DC$45_INV$950"
       + /"ctrlcount<4>" * /"$OpTx$$OpTx$FX_DC$45_INV$950" *
       FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK *
       "count<4>".LFBK * /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
       + /"ctrlcount<3>" * /"ctrlcount<4>" *
       /"$OpTx$$OpTx$FX_DC$45_INV$950" * FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK
*
       "count<3>".LFBK * /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
       + /"ctrlcount<3>" * /"$OpTx$$OpTx$FX_DC$45_INV$950" *
       FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK *
       "count<3>".LFBK * "count<4>".LFBK *
       /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
;Imported pterms FB1_3
       + /"ctrlcount<2>" * /"ctrlcount<3>" * /"ctrlcount<4>" *
       /"$OpTx$$OpTx$FX_DC$45_INV$950" * FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK
*
       "count<2>".LFBK * /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
       + /"ctrlcount<2>" * /"ctrlcount<4>" *
       /"$OpTx$$OpTx$FX_DC$45_INV$950" * "$OpTx$FX_DC$37" * FC_0_.OUT *
       /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK * "count<3>".LFBK *
       /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
       + /"ctrlcount<2>" * /"$OpTx$$OpTx$FX_DC$45_INV$950" *
       "$OpTx$FX_DC$37" * FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK *
       "count<3>".LFBK * "count<4>".LFBK *
       /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
       + /"ctrlcount<3>" * /"ctrlcount<4>" *
       /"$OpTx$$OpTx$FX_DC$45_INV$950" * "$OpTx$FX_DC$37" * FC_0_.OUT *
       /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK * "count<2>".LFBK *
       /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
       + /"ctrlcount<3>" * /"$OpTx$$OpTx$FX_DC$45_INV$950" *
       "$OpTx$FX_DC$37" * FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK *
       "count<2>".LFBK * "count<4>".LFBK *
       /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
;Imported pterms FB1_2
       + /"ctrlcount<2>" * /"ctrlcount<3>" * /"ctrlcount<4>" *
       /"$OpTx$$OpTx$FX_DC$45_INV$950" * "$OpTx$FX_DC$37" * FC_0_.OUT *
       /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK * /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
       + /"ctrlcount<2>" * /"ctrlcount<3>" *
       /"$OpTx$$OpTx$FX_DC$45_INV$950" * "$OpTx$FX_DC$37" * FC_0_.OUT *
```

```
        /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK * "count<4>".LFBK *
        /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
;Imported pterms FB1_6
        + /"ctrlcount<2>" * /"ctrlcount<3>" *
        /"$OpTx$$OpTx$FX_DC$45_INV$950" * FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK
*
        "count<2>".LFBK * "count<4>".LFBK *
        /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
        + /"ctrlcount<2>" * /"ctrlcount<4>" *
        /"$OpTx$$OpTx$FX_DC$45_INV$950" * FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK
*
        "count<2>".LFBK * "count<3>".LFBK *
        /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
        + /"ctrlcount<2>" * /"$OpTx$$OpTx$FX_DC$45_INV$950" *
        FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK *
        "count<2>".LFBK * "count<3>".LFBK * "count<4>".LFBK *
        /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
        + /"ctrlcount<4>" * /"$OpTx$$OpTx$FX_DC$45_INV$950" *
        "$OpTx$FX_DC$37" * FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK *
        "count<2>".LFBK * "count<3>".LFBK *
        /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
        + /"$OpTx$$OpTx$FX_DC$45_INV$950" * "$OpTx$FX_DC$37" *
        FC_0_.OUT * /"$OpTx$$OpTx$FX_DC$16_INV$952".LFBK *
        "count<2>".LFBK * "count<3>".LFBK * "count<4>".LFBK *
        /"$OpTx$$OpTx$FX_DC$17_INV$953".LFBK
    PWMCtrlOut.CLKF  =  CLK   ;FCLK/GCK
    PWMCtrlOut.RSTF  =  RESET ;GSR
    PWMCtrlOut.PRLD  =  GND

 START  :=  RESET * START.LFBK
        + /RESET * /"count1<1>" * /"count1<0>" * FC_3_.OUT
    START.CLKF  =  CLK ;FCLK/GCK
    START.PRLD  =  GND

"count1<0>"  :=  RESET * "count1<0>".LFBK
        + /RESET * "count<0>".LFBK
    "count1<0>".CLKF  =  CLK  ;FCLK/GCK
    "count1<0>".PRLD  =  GND

"count1<1>"  :=  ;Imported pterms FB6_2
          RESET * "count1<1>".LFBK
        + /RESET * "count<1>".LFBK
    "count1<1>".CLKF  =  CLK  ;FCLK/GCK
    "count1<1>".PRLD  =  GND

"count1<2>"  :=  RESET * "count1<2>".LFBK
        + /RESET * "count<2>".LFBK
    "count1<2>".CLKF  =  CLK  ;FCLK/GCK
    "count1<2>".PRLD  =  GND

"count1<3>"  :=  RESET * "count1<3>".LFBK
        + /RESET * "count<3>".LFBK
    "count1<3>".CLKF  =  CLK  ;FCLK/GCK
    "count1<3>".PRLD  =  GND

"count1<4>"  :=  RESET * "count1<4>".LFBK
        + /RESET * "count<4>".LFBK
    "count1<4>".CLKF  =  CLK  ;FCLK/GCK
    "count1<4>".PRLD  =  GND

"count1<5>"  :=  RESET * "count1<5>".LFBK
        + /RESET * "count<5>".LFBK
    "count1<5>".CLKF  =  CLK  ;FCLK/GCK
    "count1<5>".PRLD  =  GND

"count1<6>"  :=  RESET * "count1<6>".LFBK
        + /RESET * "count<6>".LFBK
    "count1<6>".CLKF  =  CLK  ;FCLK/GCK
```

```
    "count1<6>".PRLD  =  GND

"count1<7>"  :=  RESET * "count1<7>".LFBK
       + /RESET * "count<7>".LFBK
    "count1<7>".CLKF  =  CLK  ;FCLK/GCK
    "count1<7>".PRLD  =  GND

"count1<8>"  :=  RESET * "count1<8>".LFBK
       + /RESET * "count<8>".LFBK
    "count1<8>".CLKF  =  CLK  ;FCLK/GCK
    "count1<8>".PRLD  =  GND
```

```
"count<0>"  :=  ;Imported pterms FB5_5
         RESET * "count<0>".LFBK
       + /RESET * /"count1<0>".LFBK
    "count<0>".CLKF  =  CLK   ;FCLK/GCK
    "count<0>".PRLD  =  GND

"count<1>"  :=  RESET * "count<1>".LFBK
       + /RESET * "count1<0>" * /"count1<1>".LFBK
       + /RESET * /"count1<0>" * "count1<1>".LFBK
    "count<1>".CLKF  =  CLK   ;FCLK/GCK
    "count<1>".PRLD  =  GND

"count<2>"  :=  RESET * "count<2>".LFBK
       + /RESET * FC_1_.OUT * /"count1<2>".LFBK
       + /RESET * /FC_1_.OUT * "count1<2>".LFBK
    "count<2>".CLKF  =  CLK   ;FCLK/GCK
    "count<2>".PRLD  =  GND

"count<3>"  :=  RESET * "count<3>".LFBK
       + /RESET * /FC_1_.OUT * "count1<3>".LFBK
       + /RESET * /"count1<2>".LFBK * "count1<3>".LFBK
;Imported pterms FB1_13
       + /RESET * FC_1_.OUT * "count1<2>".LFBK *
       /"count1<3>".LFBK
    "count<3>".CLKF  =  CLK   ;FCLK/GCK
    "count<3>".PRLD  =  GND

 "count<4>"  :=  RESET * "count<4>".LFBK
       + /RESET * /"count1<2>".LFBK * "count1<4>".LFBK
       + /RESET * /"count1<3>".LFBK * "count1<4>".LFBK
;Imported pterms FB1_14
       + /RESET * /FC_1_.OUT * "count1<4>".LFBK
       + /RESET * FC_1_.OUT * "count1<2>".LFBK *
       "count1<3>".LFBK * /"count1<4>".LFBK
    "count<4>".CLKF  =  CLK   ;FCLK/GCK
    "count<4>".PRLD  =  GND

 "count<5>"  :=  RESET * "count<5>".LFBK
       + /RESET * /"count1<2>".LFBK * "count1<5>".LFBK
       + /RESET * /"count1<3>".LFBK * "count1<5>".LFBK
       + /RESET * "count1<5>".LFBK * /"count1<4>".LFBK
;Imported pterms FB1_15
       + /RESET * /FC_1_.OUT * "count1<5>".LFBK
       + /RESET * FC_1_.OUT * "count1<2>".LFBK *
       "count1<3>".LFBK * /"count1<5>".LFBK * "count1<4>".LFBK
    "count<5>".CLKF  =  CLK   ;FCLK/GCK
    "count<5>".PRLD  =  GND

 "count<6>"  :=  RESET * "count<6>".LFBK
       + /RESET * /"count1<2>".LFBK * "count1<6>".LFBK
       + /RESET * /"count1<3>".LFBK * "count1<6>".LFBK
       + /RESET * /"count1<5>".LFBK * "count1<6>".LFBK
       + /RESET * /"count1<4>".LFBK * "count1<6>".LFBK
;Imported pterms FB1_16
       + /RESET * /FC_1_.OUT * "count1<6>".LFBK
;Imported pterms FB1_18
```

```
        + /RESET * FC_1_.OUT * "count1<2>".LFBK *
        "count1<3>".LFBK * "count1<5>".LFBK * "count1<4>".LFBK *
        /"count1<6>".LFBK
    "count<6>".CLKF  =  CLK   ;FCLK/GCK
    "count<6>".PRLD  =  GND

"count<7>"  :=  RESET * "count<7>".LFBK
        + /RESET * /"count1<1>" * "count1<7>".LFBK
        + /RESET * /"count1<2>" * "count1<7>".LFBK
        + /RESET * /"count1<0>".LFBK * "count1<7>".LFBK
;Imported pterms FB5_17
        + /RESET * /"count1<3>" * "count1<7>".LFBK
        + /RESET * /"count1<5>" * "count1<7>".LFBK
        + /RESET * /"count1<4>" * "count1<7>".LFBK
        + /RESET * /"count1<6>" * "count1<7>".LFBK
        + /RESET * "count1<1>" * "count1<2>" * "count1<3>" *
        "count1<5>" * "count1<4>" * "count1<6>" * "count1<0>".LFBK *
        /"count1<7>".LFBK
    "count<7>".CLKF  =  CLK   ;FCLK/GCK
    "count<7>".PRLD  =  GND

 "count<8>"  :=  RESET * "count<8>".LFBK
;Imported pterms FB3_16
        + /RESET * /"count1<1>" * "count1<8>".LFBK
        + /RESET * /"count1<0>" * "count1<8>".LFBK
        + /RESET * /"count1<2>" * "count1<8>".LFBK
        + /RESET * /"count1<3>" * "count1<8>".LFBK
        + /RESET * /"count1<4>" * "count1<8>".LFBK
;Imported pterms FB3_15
        + /RESET * /"count1<5>" * "count1<8>".LFBK
        + /RESET * /"count1<6>" * "count1<8>".LFBK
        + /RESET * /"count1<7>" * "count1<8>".LFBK
        + /RESET * "count1<1>" * "count1<0>" * "count1<2>" *
        "count1<3>" * "count1<5>" * "count1<4>" * "count1<6>" *
        "count1<7>" * /"count1<8>".LFBK
    "count<8>".CLKF  =  CLK   ;FCLK/GCK
    "count<8>".PRLD  =  GND

/"ctrlcount1<0>".T  =  "ctrlcount1<0>".LFBK * "ctrlcount<0>".LFBK
        + /"ctrlcount1<0>".LFBK * /"ctrlcount<0>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "ctrlcount1<0>".CLKF  =  CLK     ;FCLK/GCK
    "ctrlcount1<0>".RSTF  =  RESET   ;GSR
    "ctrlcount1<0>".PRLD  =  GND

/"ctrlcount1<10>".T  =  "ctrlcount1<10>".LFBK * "ctrlcount<10>".LFBK
        + /"ctrlcount1<10>".LFBK * /"ctrlcount<10>".LFBK
        + /"count1<1>" * /"count1<2>" * /"count1<3>" *
        /"count1<5>" * /"count1<4>" * /"count1<6>" * /"count1<8>" *
        /"count1<0>".LFBK * /"count1<7>".LFBK
    "ctrlcount1<10>".CLKF  =  CLK     ;FCLK/GCK
    "ctrlcount1<10>".RSTF  =  RESET   ;GSR
    "ctrlcount1<10>".PRLD  =  GND

/"ctrlcount1<1>".T  =  "ctrlcount<1>" * "ctrlcount1<1>".LFBK
        + /"ctrlcount<1>" * /"ctrlcount1<1>".LFBK
        + /"count1<1>" * /"count1<0>" * /"count1<2>" *
        /"count1<3>" * /"count1<5>" * /"count1<4>" * /"count1<6>" *
        /"count1<7>" * /"count1<8>".LFBK
    "ctrlcount1<1>".CLKF  =  CLK     ;FCLK/GCK
    "ctrlcount1<1>".RSTF  =  RESET   ;GSR
    "ctrlcount1<1>".PRLD  =  GND

/"ctrlcount1<2>".T  =  "ctrlcount1<2>".LFBK * "ctrlcount<2>".LFBK
        + /"ctrlcount1<2>".LFBK * /"ctrlcount<2>".LFBK
        + /"count1<1>" * /"count1<0>" * /"count1<2>" *
        /"count1<3>" * /"count1<5>" * /"count1<4>" * /"count1<6>" *
        /"count1<7>" * /"count1<8>".LFBK
```

```
    "ctrlcount1<2>".CLKF  =  CLK      ;FCLK/GCK
    "ctrlcount1<2>".RSTF  =  RESET    ;GSR
    "ctrlcount1<2>".PRLD  =  GND

/"ctrlcount1<3>".T  =  "ctrlcount1<3>".LFBK * "ctrlcount<3>".LFBK
       + /"ctrlcount1<3>".LFBK * /"ctrlcount<3>".LFBK
       + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "ctrlcount1<3>".CLKF  =  CLK      ;FCLK/GCK
    "ctrlcount1<3>".RSTF  =  RESET    ;GSR
    "ctrlcount1<3>".PRLD  =  GND

/"ctrlcount1<4>".T  =  "ctrlcount1<4>".LFBK * "ctrlcount<4>".LFBK
       + /"ctrlcount1<4>".LFBK * /"ctrlcount<4>".LFBK
       + /"count1<1>" * /"count1<2>" * /"count1<3>" *
       /"count1<5>" * /"count1<4>" * /"count1<6>" * /"count1<8>" *
       /"count1<0>".LFBK * /"count1<7>".LFBK
    "ctrlcount1<4>".CLKF  =  CLK      ;FCLK/GCK
    "ctrlcount1<4>".RSTF  =  RESET    ;GSR
    "ctrlcount1<4>".PRLD  =  GND

/"ctrlcount1<5>".T  =  "ctrlcount<5>" * "ctrlcount1<5>".LFBK
       + /"ctrlcount<5>" * /"ctrlcount1<5>".LFBK
       + /"count1<1>" * /"count1<2>" * /"count1<3>" *
       /"count1<5>" * /"count1<4>" * /"count1<6>" * /"count1<8>" *
       /"count1<0>".LFBK * /"count1<7>".LFBK
    "ctrlcount1<5>".CLKF  =  CLK      ;FCLK/GCK
    "ctrlcount1<5>".RSTF  =  RESET    ;GSR
    "ctrlcount1<5>".PRLD  =  GND

/"ctrlcount1<6>".T  =  "ctrlcount1<6>".LFBK * "ctrlcount<6>".LFBK
       + /"ctrlcount1<6>".LFBK * /"ctrlcount<6>".LFBK
       + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "ctrlcount1<6>".CLKF  =  CLK      ;FCLK/GCK
    "ctrlcount1<6>".RSTF  =  RESET    ;GSR
    "ctrlcount1<6>".PRLD  =  GND

/"ctrlcount1<7>".T  =  "ctrlcount1<7>".LFBK * "ctrlcount<7>".LFBK
       + /"ctrlcount1<7>".LFBK * /"ctrlcount<7>".LFBK
       + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "ctrlcount1<7>".CLKF  =  CLK      ;FCLK/GCK
    "ctrlcount1<7>".RSTF  =  RESET    ;GSR
    "ctrlcount1<7>".PRLD  =  GND

/"ctrlcount1<8>".T  =  "ctrlcount1<8>".LFBK * "ctrlcount<8>".LFBK
       + /"ctrlcount1<8>".LFBK * /"ctrlcount<8>".LFBK
       + /"count1<1>" * /"count1<0>" * /"count1<2>" *
       /"count1<3>" * /"count1<5>" * /"count1<4>" * /"count1<6>" *
       /"count1<7>" * /"count1<8>".LFBK
    "ctrlcount1<8>".CLKF  =  CLK      ;FCLK/GCK
    "ctrlcount1<8>".SETF  =  RESET    ;GSR
    "ctrlcount1<8>".PRLD  =  GND

/"ctrlcount1<9>".T  =  "ctrlcount1<9>".LFBK * "ctrlcount<9>".LFBK
       + /"ctrlcount1<9>".LFBK * /"ctrlcount<9>".LFBK
       + /"count1<1>" * /"count1<2>" * /"count1<3>" *
       /"count1<5>" * /"count1<4>" * /"count1<6>" * /"count1<8>" *
       /"count1<0>".LFBK * /"count1<7>".LFBK
    "ctrlcount1<9>".CLKF  =  CLK      ;FCLK/GCK
    "ctrlcount1<9>".RSTF  =  RESET    ;GSR
    "ctrlcount1<9>".PRLD  =  GND

/"ctrlcount<0>"  :=  /"$OpTx$$OpTx$FX_SC$51_INV$955"
       + RESET * /"ctrlcount<0>".LFBK
       + /"$OpTx$$OpTx$FX_SC$49_INV$954" *
       "ctrlcount1<0>".LFBK * "diff1<0>".LFBK
;Imported pterms FB2_12
       + /"$OpTx$$OpTx$FX_SC$49_INV$954" *
       /"ctrlcount1<0>".LFBK * /"diff1<0>".LFBK
```

```
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT *
        /"ctrlcount<0>".LFBK
    "ctrlcount<0>".CLKF  =  CLK        ;FCLK/GCK
    "ctrlcount<0>".PRLD  =  GND

"ctrlcount<10>"  :=  "$OpTx$$OpTx$FX_SC$50_INV$956" *
        "$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount<10>".LFBK
        + "diff1<9>" * "diff1<10>" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949" *
        "ctrlcount1<9>".LFBK * "ctrlcount1<10>".LFBK
        + "diff1<9>" * /"diff1<10>" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949" *
        "ctrlcount1<9>".LFBK * /"ctrlcount1<10>".LFBK
        + /"diff1<9>" * "diff1<10>" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949" *
        /"ctrlcount1<9>".LFBK * "ctrlcount1<10>".LFBK
        + /"diff1<9>" * /"diff1<10>" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949" *
        /"ctrlcount1<9>".LFBK * "ctrlcount1<10>".LFBK
;Imported pterms FB5_1
        + "ctrlcount1<8>" * "diff1<9>" * "diff1<10>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<10>".LFBK
        + "ctrlcount1<8>" * "diff1<10>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<9>".LFBK * "ctrlcount1<10>".LFBK
        + /"ctrlcount1<8>" * /"diff1<9>" * "diff1<10>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<10>".LFBK
        + "diff1<9>" * "diff1<10>" *
        "C128_C7_C6/C128_C7_C6_D2" * /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949" *
        "ctrlcount1<10>".LFBK
        + "diff1<10>" * "C128_C7_C6/C128_C7_C6_D2" *
        /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<9>".LFBK * "ctrlcount1<10>".LFBK
;Imported pterms FB5_18
        + /"ctrlcount1<8>" * /"diff1<10>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<9>".LFBK * "ctrlcount1<10>".LFBK
;Imported pterms FB5_3
        + /"ctrlcount1<8>" * /"diff1<9>" * /"diff1<10>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<10>".LFBK
        + /"ctrlcount1<8>" * "diff1<10>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<9>".LFBK * /"ctrlcount1<10>".LFBK
        + /"diff1<9>" * "diff1<10>" *
        /"C128_C7_C6/C128_C7_C6_D2" * /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949" *
        /"ctrlcount1<10>".LFBK
        + /"diff1<9>" * /"diff1<10>" *
        /"C128_C7_C6/C128_C7_C6_D2" * /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949" *
        "ctrlcount1<10>".LFBK
        + "diff1<10>" * /"C128_C7_C6/C128_C7_C6_D2" *
        /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<9>".LFBK * /"ctrlcount1<10>".LFBK
;Imported pterms FB5_4
        + "ctrlcount1<8>" * "diff1<9>" * /"diff1<10>" *
```

```
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<10>".LFBK
        + "ctrlcount1<8>" * /"diff1<10>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<9>".LFBK * /"ctrlcount1<10>".LFBK
        + "diff1<9>" * /"diff1<10>" *
        "C128_C7_C6/C128_C7_C6_D2" * /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949" *
        /"ctrlcount1<10>".LFBK
        + /"diff1<10>" * "C128_C7_C6/C128_C7_C6_D2" *
        /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<9>".LFBK * /"ctrlcount1<10>".LFBK
        + /"diff1<10>" * /"C128_C7_C6/C128_C7_C6_D2" *
        /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<9>".LFBK * "ctrlcount1<10>".LFBK
    "ctrlcount<10>".CLKF  =  CLK      ;FCLK/GCK
    "ctrlcount<10>".PRLD  =  GND

 "ctrlcount<1>"  :=  "$OpTx$syn858/syn858_D2_INV$949" *
        "$OpTx$$OpTx$FX_SC$50_INV$956".LFBK * "ctrlcount<1>".LFBK
;Imported pterms FB4_13
        + "ctrlcount1<1>" * "diff1<1>" *
        /"$OpTx$syn858/syn858_D2_INV$949" * FC_2_.OUT *
"$OpTx$$OpTx$FX_SC$50_INV$956".LFBK
        + "ctrlcount1<1>" * /"diff1<1>" *
        /"$OpTx$syn858/syn858_D2_INV$949" * /FC_2_.OUT *
"$OpTx$$OpTx$FX_SC$50_INV$956".LFBK
        + /"ctrlcount1<1>" * "diff1<1>" *
        /"$OpTx$syn858/syn858_D2_INV$949" * /FC_2_.OUT *
"$OpTx$$OpTx$FX_SC$50_INV$956".LFBK
        + /"ctrlcount1<1>" * /"diff1<1>" *
        /"$OpTx$syn858/syn858_D2_INV$949" * FC_2_.OUT *
"$OpTx$$OpTx$FX_SC$50_INV$956".LFBK
    "ctrlcount<1>".CLKF  =  CLK      ;FCLK/GCK
    "ctrlcount<1>".PRLD  =  GND

/"ctrlcount<2>"  :=  /"$OpTx$$OpTx$FX_SC$51_INV$955"
        + RESET * /"ctrlcount<2>".LFBK
        + /"ctrlcount1<0>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"ctrlcount1<1>".LFBK * "ctrlcount1<2>".LFBK * "diff1<2>".LFBK
        + /"ctrlcount1<0>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "ctrlcount1<2>".LFBK * "diff1<2>".LFBK * /"diff1<1>".LFBK
        + /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"ctrlcount1<1>".LFBK * "ctrlcount1<2>".LFBK * "diff1<2>".LFBK *
        /"diff1<1>".LFBK
;Imported pterms FB3_2
        + /"ctrlcount1<0>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"ctrlcount1<1>".LFBK * /"ctrlcount1<2>".LFBK * /"diff1<2>".LFBK
        + /"ctrlcount1<0>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"ctrlcount1<2>".LFBK * /"diff1<2>".LFBK * /"diff1<1>".LFBK
        + /"diff1<0>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"ctrlcount1<1>".LFBK * "ctrlcount1<2>".LFBK * "diff1<2>".LFBK
        + /"diff1<0>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "diff1<1>".LFBK
;Imported pterms FB3_18
        + /"diff1<0>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"ctrlcount1<1>".LFBK * "ctrlcount1<2>".LFBK * "diff1<2>".LFBK
        + /"diff1<0>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "ctrlcount1<2>".LFBK * "diff1<2>".LFBK * /"diff1<1>".LFBK
        + /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "ctrlcount1<1>".LFBK * "ctrlcount1<2>".LFBK * /"diff1<2>".LFBK *
```

```
        "diff1<1>".LFBK
        + /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "ctrlcount1<1>".LFBK * /"ctrlcount1<2>".LFBK * "diff1<2>".LFBK *
        "diff1<1>".LFBK
        + /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"ctrlcount1<1>".LFBK * /"ctrlcount1<2>".LFBK * /"diff1<2>".LFBK *
        /"diff1<1>".LFBK
;Imported pterms FB3_17
        + "ctrlcount1<0>" * "diff1<0>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" * "ctrlcount1<1>".LFBK * "ctrlcount1<2>".LFBK *
        /"diff1<2>".LFBK
        + "ctrlcount1<0>" * "diff1<0>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" * "ctrlcount1<1>".LFBK * /"ctrlcount1<2>".LFBK *
        "diff1<2>".LFBK
        + "ctrlcount1<0>" * "diff1<0>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" * /"ctrlcount1<2>".LFBK * "diff1<2>".LFBK *
        "diff1<1>".LFBK
        + /"count1<1>" * /"count1<0>" * /"count1<2>" *
        /"count1<3>" * /"count1<5>" * /"count1<4>" * /"count1<6>" *
        /"count1<7>" * /"count1<8>".LFBK * /"ctrlcount<2>".LFBK
    "ctrlcount<2>".CLKF  =  CLK       ;FCLK/GCK
    "ctrlcount<2>".PRLD  =  GND

/"ctrlcount<3>".T  =  RESET * "$OpTx$$OpTx$FX_SC$50_INV$956"
        + /"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"ctrlcount<3>".LFBK
        + "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "ctrlcount1<3>".LFBK * /"ctrlcount<3>".LFBK
        + /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"ctrlcount1<3>".LFBK * /"ctrlcount<3>".LFBK
        + /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * "ctrlcount1<3>".LFBK * "ctrlcount<3>".LFBK
;Imported pterms FB2_13
        + /"count1<1>" * /"count1<0>" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * FC_3_.OUT
        + "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"ctrlcount1<3>".LFBK * "ctrlcount<3>".LFBK
    "ctrlcount<3>".CLKF  =  CLK       ;FCLK/GCK
    "ctrlcount<3>".PRLD  =  GND

 "ctrlcount<4>"  :=  "$OpTx$$OpTx$FX_SC$50_INV$956" *
        "$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount<4>".LFBK
        + "ctrlcount1<3>" * "diff1<4>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<4>".LFBK
        + "ctrlcount1<3>" * /"diff1<4>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<4>".LFBK
        + /"ctrlcount1<3>" * "diff1<4>" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * "ctrlcount1<3>".LFBK * "ctrlcount<3>".LFBK
*
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<4>".LFBK
        + /"ctrlcount1<3>" * /"diff1<4>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<4>".LFBK
;Imported pterms FB5_15
        + "diff1<4>" * "C128_C2_C6/C128_C2_C6_D2" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<4>".LFBK
        + "diff1<4>" * /"C128_C2_C6/C128_C2_C6_D2" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<4>".LFBK
        + /"diff1<4>" * "C128_C2_C6/C128_C2_C6_D2" *
```

```
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<4>".LFBK
        + /"diff1<4>" * /"C128_C2_C6/C128_C2_C6_D2" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<4>".LFBK
    "ctrlcount<4>".CLKF  =  CLK       ;FCLK/GCK
    "ctrlcount<4>".PRLD  =  GND

 "ctrlcount<5>"  :=  "$OpTx$$OpTx$FX_SC$50_INV$956" *
        "$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount<5>".LFBK
        + "ctrlcount1<4>" * "ctrlcount1<5>" * "diff1<4>" *
        "diff1<5>" * "$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949"
        + "ctrlcount1<4>" * /"ctrlcount1<5>" * "diff1<4>" *
        /"diff1<5>" * "$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949"
        + /"ctrlcount1<4>" * "ctrlcount1<5>" * /"diff1<4>" *
        /"diff1<5>" * "$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949"
        + /"ctrlcount1<4>" * /"ctrlcount1<5>" * /"diff1<4>" *
        "diff1<5>" * "$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949"
;Imported pterms FB6_6
        + "ctrlcount1<3>" * "ctrlcount1<4>" *
        "ctrlcount1<5>" * "diff1<5>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949"
        + "ctrlcount1<3>" * "ctrlcount1<5>" * "diff1<4>" *
        "diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + /"ctrlcount1<3>" * /"ctrlcount1<5>" * /"diff1<4>" *
        "diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + "ctrlcount1<5>" * "diff1<4>" * "diff1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
;Imported pterms FB6_5
        + /"ctrlcount1<3>" * /"ctrlcount1<4>" *
        "ctrlcount1<5>" * /"diff1<5>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949"
;Imported pterms FB6_8
        + /"ctrlcount1<3>" * /"ctrlcount1<4>" *
        /"ctrlcount1<5>" * "diff1<5>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949"
        + /"ctrlcount1<3>" * "ctrlcount1<5>" * /"diff1<4>" *
        /"diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + /"ctrlcount1<4>" * /"ctrlcount1<5>" * "diff1<5>" *
        /"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + "ctrlcount1<5>" * /"diff1<4>" * /"diff1<5>" *
        /"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + /"ctrlcount1<5>" * /"diff1<4>" * "diff1<5>" *
        /"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
;Imported pterms FB6_9
        + "ctrlcount1<3>" * "ctrlcount1<4>" *
```

```
        /"ctrlcount1<5>" * /"diff1<5>" *
        "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949"
        + "ctrlcount1<3>" * /"ctrlcount1<5>" * "diff1<4>" *
        /"diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + "ctrlcount1<4>" * /"ctrlcount1<5>" * /"diff1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + /"ctrlcount1<4>" * "ctrlcount1<5>" * /"diff1<5>" *
        /"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + /"ctrlcount1<4>" * "ctrlcount1<5>" * /"diff1<5>" *
        /"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + "ctrlcount1<5>" * /"ctrlcount1<5>" * "diff1<4>" *
        /"diff1<5>" * "C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + "ctrlcount1<4>" * /"ctrlcount1<5>" * /"diff1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + /"ctrlcount1<4>" * "ctrlcount1<5>" * /"diff1<5>" *
        /"C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
        + /"ctrlcount1<5>" * "diff1<4>" * /"diff1<5>" *
        "C128_C2_C6/C128_C2_C6_D2" * /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        "$OpTx$$OpTx$FX_SC$50_INV$956" * /"$OpTx$syn858/syn858_D2_INV$949"
    "ctrlcount<5>".CLKF  =  CLK       ;FCLK/GCK
    "ctrlcount<5>".PRLD  =  GND

 "ctrlcount<6>"  :=  /"$OpTx$$OpTx$FX_SC$51_INV$955"
        + RESET * "ctrlcount<6>".LFBK
        + /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * "ctrlcount1<6>".LFBK
;Imported pterms FB2_11
        + /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * /"ctrlcount1<6>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT *
        "ctrlcount<6>".LFBK
    "ctrlcount<6>".CLKF  =  CLK       ;FCLK/GCK
    "ctrlcount<6>".PRLD  =  GND

 "ctrlcount<7>"  :=  /"$OpTx$$OpTx$FX_SC$51_INV$955"
        + RESET * "ctrlcount<7>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT *
        "ctrlcount<7>".LFBK
        + "diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * "ctrlcount1<6>".LFBK *
"ctrlcount1<7>".LFBK
        + /"diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * "ctrlcount1<6>".LFBK *
/"ctrlcount1<7>".LFBK
;Imported pterms FB2_1
        + /"ctrlcount1<5>" * "diff1<7>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
        /"ctrlcount1<7>".LFBK *
        "C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + "ctrlcount1<4>" * "diff1<7>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
        "C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" * "ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + /"ctrlcount1<4>" * /"diff1<7>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
        "C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" * /"ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + /"ctrlcount1<4>" * "diff1<7>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
        "C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" * /"ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + /"ctrlcount1<4>" * /"diff1<7>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
        "C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" * "ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
;Imported pterms FB2_2
```

```
        + "diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * /"C128_C2_C6/C128_C2_C6_D2" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" *
        /"ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + /"diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * "C128_C2_C6/C128_C2_C6_D2" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" *
        /"ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + /"diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * /"C128_C2_C6/C128_C2_C6_D2" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" *
        /"ctrlcount1<7>".LFBK *
        "C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + "diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * /"C128_C2_C6/C128_C2_C6_D2" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" *
        "ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
;Imported pterms FB2_17
        + "diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * /"ctrlcount1<6>".LFBK *
/"ctrlcount1<7>".LFBK
        + /"diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * /"ctrlcount1<6>".LFBK *
"ctrlcount1<7>".LFBK
        + "ctrlcount1<5>" * "diff1<7>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
        "ctrlcount1<7>".LFBK *
        "C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + "ctrlcount1<5>" * /"diff1<7>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
        /"ctrlcount1<7>".LFBK *
        "C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + /"ctrlcount1<5>" * /"diff1<7>" *
        /"$OpTx$$OpTx$FX_SC$49_INV$954" *
/"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
        "ctrlcount1<7>".LFBK *
        "C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
;Imported pterms FB2_16
        + "diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" * "C128_C2_C6/C128_C2_C6_D2" *
        /"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
/"C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" *
        "ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + "diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" * "ctrlcount1<3>".LFBK *
"ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + "diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" * /"ctrlcount1<3>".LFBK *
/"ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + /"diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
        /"C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" * "ctrlcount1<3>".LFBK *
/"ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
        + /"diff1<7>" * /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C5_C6$X0$diff1<6>/C128_C5_C6$X0$diff1<6>_D" *
"C128_C2_C6$X0$diff1<3>/C128_C2_C6$X0$diff1<3>_D" *
```

```
        /"C128_C3_C6$X0$diff1<4>/C128_C3_C6$X0$diff1<4>_D" * /"ctrlcount1<3>".LFBK *
"ctrlcount1<7>".LFBK *
        /"C128_C4_C6$X0$diff1<5>/C128_C4_C6$X0$diff1<5>_D".LFBK
    "ctrlcount<7>".CLKF  =  CLK       ;FCLK/GCK
    "ctrlcount<7>".PRLD  =  GND

 "ctrlcount<8>"  :=  /"$OpTx$$OpTx$FX_SC$51_INV$955"
        + RESET * "ctrlcount<8>".LFBK
        + /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "ctrlcount1<8>".LFBK
;Imported pterms FB3_12
        + /"$OpTx$$OpTx$FX_SC$49_INV$954" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * /"ctrlcount1<8>".LFBK
        + /"count1<1>" * /"count1<0>" * /"count1<2>" *
        /"count1<3>" * /"count1<5>" * /"count1<4>" * /"count1<6>" *
        /"count1<7>" * /"count1<8>".LFBK * "ctrlcount<8>".LFBK
    "ctrlcount<8>".CLKF  =  CLK       ;FCLK/GCK
    "ctrlcount<8>".PRLD  =  GND

 "ctrlcount<9>"  :=  "$OpTx$$OpTx$FX_SC$50_INV$956" *
        "$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount<9>".LFBK
        + "ctrlcount1<8>" * "diff1<9>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<9>".LFBK
        + "ctrlcount1<8>" * /"diff1<9>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<9>".LFBK
        + /"ctrlcount1<8>" * "diff1<9>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<9>".LFBK
        + /"ctrlcount1<8>" * /"diff1<9>" *
        "C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" * "$OpTx$$OpTx$FX_SC$50_INV$956"
*
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<9>".LFBK
;Imported pterms FB5_13
        + "diff1<9>" * "C128_C7_C6/C128_C7_C6_D2" *
        /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<9>".LFBK
        + "diff1<9>" * /"C128_C7_C6/C128_C7_C6_D2" *
        /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * /"ctrlcount1<9>".LFBK
        + /"diff1<9>" * "C128_C7_C6/C128_C7_C6_D2" *
        /"C128_C7_C6$X0$diff1<8>/C128_C7_C6$X0$diff1<8>_D" *
"$OpTx$$OpTx$FX_SC$50_INV$956" *
        /"$OpTx$syn858/syn858_D2_INV$949" * "ctrlcount1<9>".LFBK
    "ctrlcount<9>".CLKF  =  CLK       ;FCLK/GCK
    "ctrlcount<9>".PRLD  =  GND

 "diff1<0>"  :=  RESET * "diff1<0>".LFBK
;Imported pterms FB2_7
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT *
        "diff1<0>".LFBK
    "diff1<0>".CLKF  =  CLK   ;FCLK/GCK
    "diff1<0>".PRLD  =  GND

 "diff1<10>"  :=  RESET * "diff1<10>".LFBK
;Imported pterms FB4_16
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT *
        "diff1<10>".LFBK
```

```
    "diff1<10>".CLKF  =  CLK  ;FCLK/GCK
    "diff1<10>".PRLD  =  GND

/"diff1<1>".T  =  RESET
        + "diff1<1>".LFBK * "diff<0>".LFBK
        + /"diff1<1>".LFBK * /"diff<0>".LFBK
;Imported pterms FB3_11
        + /"count1<1>" * /"count1<0>" * /"count1<2>" *
        /"count1<3>" * /"count1<5>" * /"count1<4>" * /"count1<6>" *
        /"count1<7>" * /"count1<8>".LFBK
    "diff1<1>".CLKF  =  CLK   ;FCLK/GCK
    "diff1<1>".PRLD  =  GND

/"diff1<2>".T  =  RESET
        + "diff1<2>".LFBK * "diff<1>".LFBK
        + /"diff1<2>".LFBK * /"diff<1>".LFBK
        + /"count1<1>" * /"count1<0>" * /"count1<2>" *
        /"count1<3>" * /"count1<5>" * /"count1<4>" * /"count1<6>" *
        /"count1<7>" * /"count1<8>".LFBK
    "diff1<2>".CLKF  =  CLK   ;FCLK/GCK
    "diff1<2>".PRLD  =  GND

/"diff1<3>".T  =  RESET
;Imported pterms FB4_12
        + "diff<2>".LFBK * "diff1<3>".LFBK
        + /"diff<2>".LFBK * /"diff1<3>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff1<3>".CLKF  =  CLK   ;FCLK/GCK
    "diff1<3>".PRLD  =  GND

/"diff1<4>".T  =  RESET
        + "diff1<4>".LFBK * "diff<3>".LFBK
;Imported pterms FB4_11
        + /"diff1<4>".LFBK * /"diff<3>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff1<4>".CLKF  =  CLK   ;FCLK/GCK
    "diff1<4>".PRLD  =  GND

/"diff1<5>".T  =  RESET
        + "diff1<5>".LFBK * "diff<4>".LFBK
        + /"diff1<5>".LFBK * /"diff<4>".LFBK
;Imported pterms FB4_10
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff1<5>".CLKF  =  CLK   ;FCLK/GCK
    "diff1<5>".PRLD  =  GND

/"diff1<6>".T  =  RESET
        + "diff1<6>".LFBK * "diff<5>".LFBK
        + /"diff1<6>".LFBK * /"diff<5>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff1<6>".CLKF  =  CLK   ;FCLK/GCK
    "diff1<6>".PRLD  =  GND

/"diff1<7>".T  =  RESET
        + "diff<6>" * "diff1<7>".LFBK
        + /"diff<6>" * /"diff1<7>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff1<7>".CLKF  =  CLK   ;FCLK/GCK
    "diff1<7>".PRLD  =  GND

/"diff1<8>".T  =  RESET
        + "diff<7>" * "diff1<8>".LFBK
        + /"diff<7>" * /"diff1<8>".LFBK
;Imported pterms FB4_9
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff1<8>".CLKF  =  CLK   ;FCLK/GCK
    "diff1<8>".PRLD  =  GND
```

```
/"diff1<9>".T  =  RESET
        + "diff<8>" * "diff1<9>".LFBK
;Imported pterms FB4_8
        + /"diff<8>" * /"diff1<9>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff1<9>".CLKF  =  CLK   ;FCLK/GCK
    "diff1<9>".PRLD  =  GND

/"diff<0>".T  =  RESET
        + "Vdc<0>" * /"diff<0>".LFBK
        + /"Vdc<0>" * "diff<0>".LFBK
        + /"count1<1>" * /"count1<0>" * /"count1<2>" *
        /"count1<3>" * /"count1<5>" * /"count1<4>" * /"count1<6>" *
        /"count1<7>" * /"count1<8>".LFBK
    "diff<0>".CLKF  =  CLK    ;FCLK/GCK
    "diff<0>".PRLD  =  GND

/"diff<1>".T  =  RESET
        + "Vdc<1>" * /"diff<1>".LFBK
        + /"Vdc<1>" * "diff<1>".LFBK
        + /"count1<1>" * /"count1<0>" * /"count1<2>" *
        /"count1<3>" * /"count1<5>" * /"count1<4>" * /"count1<6>" *
        /"count1<7>" * /"count1<8>".LFBK
    "diff<1>".CLKF  =  CLK    ;FCLK/GCK
    "diff<1>".PRLD  =  GND

/"diff<2>".T  =  RESET
;Imported pterms FB4_7
        + "Vdc<2>" * /"diff<2>".LFBK
        + /"Vdc<2>" * "diff<2>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff<2>".CLKF  =  CLK    ;FCLK/GCK
    "diff<2>".PRLD  =  GND

/"diff<3>".T  =  RESET
        + "Vdc<3>" * /"diff<3>".LFBK
        + /"Vdc<3>" * "diff<3>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff<3>".CLKF  =  CLK    ;FCLK/GCK
    "diff<3>".PRLD  =  GND

/"diff<4>".T  =  RESET
        + "Vdc<4>" * /"diff<4>".LFBK
        + /"Vdc<4>" * "diff<4>".LFBK
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff<4>".CLKF  =  CLK    ;FCLK/GCK
    "diff<4>".PRLD  =  GND

/"diff<5>".T  =  RESET
        + "Vdc<5>" * /"diff<5>".LFBK
        + /"Vdc<5>" * "diff<5>".LFBK
;Imported pterms FB4_3
        + /"count1<1>" * /"count1<0>" * FC_3_.OUT
    "diff<5>".CLKF  =  CLK    ;FCLK/GCK
    "diff<5>".PRLD  =  GND

/"diff<6>".T  =  RESET
;Imported pterms FB5_12
        + "Vdc<6>" * /"diff<6>".LFBK
        + /"Vdc<6>" * "diff<6>".LFBK
        + /"count1<1>" * /"count1<2>" * /"count1<3>" *
        /"count1<5>" * /"count1<4>" * /"count1<6>" * /"count1<8>" *
        /"count1<0>".LFBK * /"count1<7>".LFBK
    "diff<6>".CLKF  =  CLK    ;FCLK/GCK
    "diff<6>".PRLD  =  GND

/"diff<7>".T  =  RESET
        + "Vdc<7>" * "diff<7>".LFBK
```

```
;Imported pterms FB5_11
        + /"Vdc<7>" * /"diff<7>".LFBK
        + /"count1<1>" * /"count1<2>" * /"count1<3>" *
        /"count1<5>" * /"count1<4>" * /"count1<6>" * /"count1<8>" *
        /"count1<0>".LFBK * /"count1<7>".LFBK
    "diff<7>".CLKF  =  CLK    ;FCLK/GCK
    "diff<7>".PRLD  =  GND

/"diff<8>".T  =  RESET
        + "Vdc<7>" * "diff<8>".LFBK
        + /"Vdc<7>" * /"diff<8>".LFBK
;Imported pterms FB5_10
        + /"count1<1>" * /"count1<2>" * /"count1<3>" *
        /"count1<5>" * /"count1<4>" * /"count1<6>" * /"count1<8>" *
        /"count1<0>".LFBK * /"count1<7>".LFBK
    "diff<8>".CLKF  =  CLK    ;FCLK/GCK
    "diff<8>".PRLD  =  GND
```

Following is a list of all global compiler options used by the fitter run.

```
Device(s) Specified                          : XC95108-7-PC84
Use Timing Constraints                       : ON
Ignore Assignments In Design File            : OFF
Create Programmable Ground Pins              : OFF
Use Advanced Fitting                         : ON
Use Local Feedback                           : ON
Use Pin Feedback                             : ON
Default Power Setting                        : STD
Default Output Slew Rate                     : FAST
Guide File Used                              : NONE
Multi Level Logic Optimization               : ON
Timing Optimization                          : ON
Power/Slew Optimization                      : OFF
High Fitting Effort                          : ON
Automatic Wire-ANDing                        : ON
Xor Synthesis                                : ON
D/T Synthesis                                : ON
Use Boolean Minimization                     : ON
Global Clock(GCK) Optimization               : ON
Global Set/Reset(GSR) Optimization           : ON
Global Output Enable(GTS) Optimization       : ON
Collapsing pterm limit                       : 25
Collapsing input limit                       : 36
```